

Дэйв Ши, Молли Е. Хольцшлаг

Школа Web-мастерства

- Поражающие воображение примеры Web-дизайна на основе CSS
- Советы по использованию типографских эффектов для оживления дизайна
- Изобретательные трюки и приемы кодирования таблиц стилей, готовых встретить будущее
- Секреты CSS от ведущих дизайнеров и разработчиков
- Эффективное использование графики для ускорения загрузки Web-страниц

ФИЛОСОФИЯ CSS-ДИЗАЙНА

WWW.CSSZENGARDEN.COM

NT PRESS
New Riders

Dave Shea
Molly E. Holzschlag

The Zen of CSS Design:

Visual Enlightenment for the Web



Дэйв Ши
Молли Е. Хольцшлаг

Философия CSS-дизайна

Дэйв Ши
Молли Е. Хольцшлаг

Философия CSS-дизайна

Школа Web-мастерства

NT Press
Москва, 2005

УДК 004.738.5
ББК 32.973.26-018.2
Ш55

Подписано в печать с готовых диапозитивов 27.07.2005. Формат 84×108[1]/[16]. Бумага офсетная. Гарнитура «Миниатюра». Печать офсетная. Усл. печ. л. 32,76. Тираж 3000 экз. Заказ 1942.

Ши Д., Хольцшлаг М. Е.

Ш55 Философия CSS-дизайна / Дэйв Ши, Молли Е. Хольцшлаг ; пер. с англ. А. А. Слинкина. – М. : НТ Пресс, 2005. – 312 с. : ил. – (Школа Web-мастерства).

ISBN 5-477-00122-4

В этой книге на примере широко известного сайта CSS Zen Garden рассматривается применение каскадных таблиц стилей. Здесь нашли отражение такие вопросы, как подготовка HTML-разметки для последующего применения стилей, синтаксис языка CSS, принципы верстки, методы работы с графикой, особенности использования цвета, типографские эффекты и многое другое. Не осталась без внимания проблема неодинакового уровня поддержки CSS разными браузерами; авторы рассказывают о приемах, с помощью которых можно добиться межбраузерной совместимости.

Эта книга не является учебником по CSS, но представляет несомненный интерес для различных категорий пользователей: от новичков до маститых профессионалов.

Издание состоит из 7 глав, предметного указателя и приложения.

УДК 004.738.5
ББК 32.973.26-018.2

ISBN 0-321-30347-4 (англ.)
ISBN 5-477-00122-4 (рус.)
ISBN 985-13-4866-X (Харвест)

Об авторах

Дэйв Ши



Дэйв Ши занимается графическим дизайном, он создатель и идейный вдохновитель сайта CSS Zen Garden, который завоевал различные награды, в том числе приз Best of Show на конференции South by Southwest Interactive, проходившей в Остине, штат Техас. Дэйв часто выступает с докладами на конференциях по всему миру, а его работы украшают многочисленные книги и журналы.

Он также является членом проекта по стандартизации Web (Web Standards Project – WaSP) – группы разработчиков и дизайнеров, пропагандирующих кроссплатформенные, непатентованные технологии Web. Дэйв – владелец и директор находящейся в Ванкувере компании Bright Creative, которая занимается веб-дизайном, он публикует материалы о различных аспектах Web на сайте mezzoblue.com. В свободное время его можно встретить на фермерских рынках в Ванкувере или в какой-нибудь независимой компании по обжариванию кофейных зерен.

Молли Е. Хольцшлаг



Автор книг, преподаватель и веб-дизайнер, Молли Е. Хольцшлаг написала свыше 30 книг, посвященных веб-дизайну. Она входит в число 25 самых влиятельных женщин в Web. Без сомнения, в мире дизайна и разработки Web Молли – одна из самых ярких и энергичных личностей.

Как член оргкомитета проекта Web Standards Project (WaSP), Молли, наряду с другими преданными своему делу разработчиками и дизайнерами, всячески пропагандирует рекомендации консорциума W3C. Она также читает курсы для веб-мастеров в университете штата Аризона, университете Феникса и колледже Pima Community College. Еще на протяжении последних трех лет она вела очень популярную колонку Integrated Design в журнале Web Techniques Magazine, посвященном техническим приемам, применяемым в Web. Кроме того, один год она была исполнительным редактором сайта WebReview.com. Подробнее о Молли можно прочитать на ее сайте http://molly.com.

Оглавление



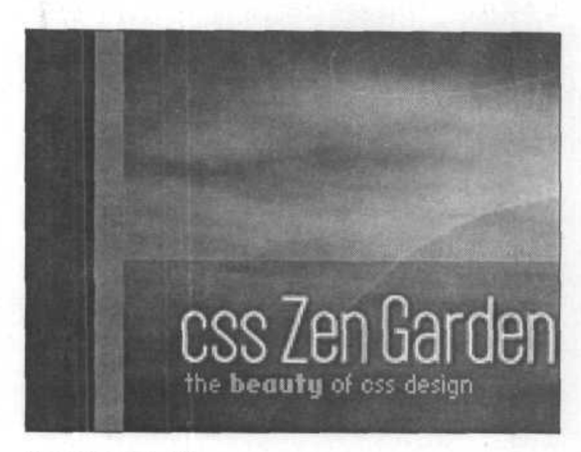
Ночная поездка, с. 84



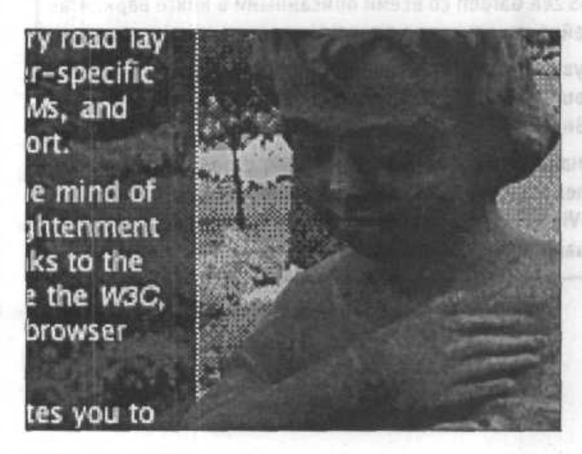
Белая лилия, с. 104



Береговой бриз, с. 158



Два океана, с. 174



Удвоение, с. 222



К югу от границы, с. 266

Введение

Добро пожаловать в книгу о проекте, который призван сформировать новое представление о Всемирной паутине.

Своим появлением эта книга обязана сайту, запущенному 8 мая 2003 года. Он изменил существовавшее до тех пор отношение к веб-дизайну и через некоторое время получил международное признание. Сайт, названный CSS Zen Garden, является результатом совместной работы многих одаренных людей из разных стран.

Сайт Zen Garden одновременно является и манифестом, и галереей; он был создан, чтобы продемонстрировать, чего можно достичь за счет применения каскадных таблиц стилей. Мы призываем всех дизайнеров предлагать собственные оригинальные работы в виде таблиц стилей и графических изображений. Но есть одно условие: базовый HTML-файл должен оставаться одним и тем же. У этого правила нет исключений; в каждом варианте дизайна HTML-разметка идентична.

Звучит неправдоподобно? Может, даже бредово? Возможно, но совершенно очевидно, что в умелых руках CSS ничем не уступает другим инструментам проектирования. В этой книге мы представим 36 лучших работ, выставленных на сайте, и подробно опишем, как их авторы подходили к проектированию и какие при этом использовались инструменты и методы кодирования.

Целая книга об одном веб-сайте? Серьезно? Если бы это был рядовой сайт, ваше недоумение было бы оправдано. Но Zen Garden содержит столько поучительных идей, столько приемов, предложенных разными людьми со всего света, что перед нами встала проблема, как все это уместить в одной книге.

О чем эта книга

Можно ли назвать эту книгу документацией и словесным описанием сайта? Отчасти. Может ли она служить руководством по заслуживающим доверия методам и приемам веб-дизайна? Определенно, да. Является ли она галереей, где можно почерпнуть идеи для собственных проектов? Без сомнения.

ПРИМЕЧАНИЕ

Сайт CSS Zen Garden со всеми описанными в книге вариантами дизайна находится по адресу www.csszengarden.com.

Чтобы увидеть разметку, найдите в своем браузере функцию **View Source** (Текст HTML), которая обычно находится в меню **View** (Вид).

Для ознакомления с таблицей стилей, с помощью которой оформлен тот или иной вариант дизайна, найдите на странице ссылку View This Design's CSS. Но будьте внимательны: для разных дизайнов местонахождение этой ссылки различно.

При работе над этой книгой нам было ясно, что в сайте Zen Garden заключено так много разнообразных знаний о современном веб-дизайне, что любая книга, посвященная какому-нибудь одному его аспекту, не смогла бы исчерпать всей глубины. Мы понимали, что необходимо раскрыть все особенности сайта. Типография, теория цвета, техника размещения элементов – в этой книге есть все.

Можно ли считать эту книгу учебником по CSS? Да, но не для начинающих. Чтобы сосредоточиться на тонкостях визуализации, нам пришлось пропустить начальную школу и поступить сразу в среднюю. Но в магазинах и так можно найти замечательные книги, посвященные основам синтаксиса и грамматики CSS. Мы рекомендуем купить какую-нибудь из них и читать ее параллельно с нашей. Тогда вы увидите, как теория претворяется в визуально привлекательные образы.

Технические приемы, описываемые в этой книге, призваны показать всю картину и проиллюстрировать способы решения задач визуализации, не останавливаясь на каждом аспекте спецификации CSS. Там, где необходимо, мы упоминаем о поддержке CSS в разных браузерах, чтобы не отрываться от реальности.

Кому предназначена эта книга

Мы писали нашу книгу для всех, кто интересуется дизайном и разработкой сайтов.

Дизайнеры

Применяя визуальный вокабулярий для обсуждения вопросов теории и практики применения CSS, мы хотели показать, как с помощью технологий, основанных на самых современных стандартах, можно решать реально возникающие задачи.

Программисты

Основной упор сделан на визуальных аспектах, но это не значит, что вы должны обладать ученой степенью в области графического дизайна. Мы старались определять все основные термины в самой книге, так что даже читатели, не знакомые с дизайном, смогут извлечь пользу из описываемых идей и методов.

Начинающие пользователи CSS

Употребление жаргона и техницизмов сведено к минимуму, поэтому большая часть материала доступна даже тем, кто имеет лишь смутное представление о технологии CSS. Мы рекомендуем прочитать также какую-нибудь книгу из рекомендуемого списка литературы, приведенного в разделе «В заключение», чтобы освоиться с синтаксисом и основными понятиями, а затем уже приступить к чтению этой книги, которая покажет, как эти идеи можно применить на практике, в том числе и к вашим собственным проектам.

В числе рассматриваемых тем есть и базовые принципы веб-дизайна, применимые в самых разных, хотя и не во всех, ситуациях. Создать веб-страницу, включающую таблицы CSS, можно и с помощью автоматических инструментов типа Macromedia Dreamweaver, но для реализации рассматриваемых нами приемов все же лучше подходит ручное кодирование в каком-нибудь HTML-редакторе. Если вы привыкли к перетаскиванию иконок с панели инструментов в рабочую область, то в некоторых отношениях эта книга может показаться технически сложной. Впрочем, мы надеемся, что это не отвратит вас от искушения обогатить свои знания.

Опытные пользователи CSS

Читатели, уже знакомые со всеми тонкостями модели прямоугольных областей, плавающим размещением и селекторами, тем не менее найдут для себя много полезного. Упор на визуальные аспекты в сочетании с примерами практического применения CSS, предложенными самыми творческими и изобретательными дизайнерами современности, – гарантия того, что даже маститые профессионалы найдут среди множества рассматриваемых тем немало для себя нового.

Принятые соглашения

Каскадные таблицы стилей (CSS) – это лишь одна из широкого спектра технологий, одобренных консорциумом World Wide Web Consortium (W3C), получивших общее название «стандарты Web». В их число входят также HTML, XHTML и XML. Когда говорят о проектировании на основе стандартов, имеют в виду весь набор. Если веб-страница составлена в соответствии со стандартами, то говорят, что она *совместима со стандартами.*

У языков HTML и XHTML много общего. Поэтому все сказанное об HTML справедливо и для XHTML, если только явно не оговорено противное.

Примечания и советы на полях относятся к тексту, расположенному рядом.

Названия свойств, определенных в CSS, и их значения печатаются моноширинным шрифтом, например: `12px`, `font-size` или `#summary`. Приведем краткий

перечень того, что имеется в виду при упоминании Internet Explorer:

Internet Explorer – все основные версии. К ним относятся версии 5.0, 5.5 и 6.0 для Windows, а также 5.2 для Mac OS X. Версии младше 5.0 сегодня уже практически не используются, так что в большинстве случаев их можно не принимать во внимание;

Internet Explorer для Windows – все основные версии для Windows. Версии для Mac не рассматриваются;

Internet Explorer для Mac – версия 5.2 для Mac OS X и, как правило, версия 5.1 *для* Mac OS.9. *Версии для* Windows не рассматриваются;

Internet Explorer x.x – если упомянут конкретный номер версии или несколько номеров, то остальные версии не рассматриваются.

Как пользоваться этой книгой

У типичного сайта имеется отправная точка, которую часто называют *начальной* или *домашней страницей*. Из нее можно попасть в любое место сайта, выбрав один из основных разделов и следуя по ссылкам внутри представленного в нем контента. Для поиска нужного контента можно также воспользоваться поисковой системой. А можно щелкнуть по какой-нибудь ссылке наугад и посмотреть, куда она приведет.

Именно такую модель мы и выбрали для данной книги. Нельзя сказать, что какой-то определенный способ работы с ней правилен или неправилен. Вы можете начать с первой страницы и читать от корки до корки, а можете сразу перейти к главе, которая вас особенно интересует. Книгу можно читать в любом порядке, начав с любого приглянувшегося варианта дизайна.

Книга состоит из двух крупных частей:

- в первой главе обсуждается устройство сайта Zen Garden и некоторые основополагающие темы, например структура разметки и методика проектирования, обеспечивающая гибкость модификаций;
- вторая часть, занимающая почти всю книгу, состоит из шести глав. В каждой рассматривается по шесть вариантов дизайна Zen Garden с точки зрения какой-либо общей темы, например типографских возможностей. Изучив различные проблемы и методы их решения на этих 36 примерах, вы освоите основные принципы веб-дизайна, а также способы применения форматирования на базе CSS.

Вот краткий обзор всех семи глав.

Глава 1. Исходный текст

Рассказ о возникновении идеи сайта CSS Zen Garden и об эволюции применяемых в нем таблиц стилей. Обсуждается структура разметки и общие принципы дизайна.

Глава 2. Дизайн

Рассматриваются основные элементы дизайна и их применение в контексте Web. Затрагиваются вопросы теории цвета, пропорций и позиционирования, отношения между шрифтами и фотографиями, а также совместное и гармоничное использование Adobe Photoshop и CSS.

Глава 3. Верстка

Создание сложных макетов страниц с помощью CSS. Верстка с несколькими колонками, на базе плавающих блоков и позиционирования.

Глава 4. Графика

Применение графики в дополнение к макету, методы создания графических изображений. Замена графики, форматы графических файлов. Где найти исходный графический материал.

Глава 5. Типография

Все, относящееся к типографии; ограничения на использование шрифтов в Web и как их обойти. Выбор размера и гарнитуры шрифта, графические шрифты.

Глава 6. Специальные эффекты

Нетривиальное применение CSS для фильтрации стилей в зависимости от браузера, создания новых гибких способов верстки и обхода технических ограничений с помощью остроумно написанного кода.

Глава 7. Реконструкция

Постойте за спиной шести дизайнеров и посмотрите, как они это делают. Мы возвращаемся к некоторым из рассмотренных ранее вариантов дизайнов и детально обсуждаем каждый шаг. На ваших глазах принципы, изложенных в предыдущих пяти главах, применяются на практике и дают потрясающий эффект.

Как видите, мы постарались втиснуть в эту книгу максимум возможного для того, чтобы сделать ее самым полным источником по современному веб-дизайну, основанному на стандартах.

Надеемся, что читать книгу вам будет так же приятно, как нам было приятно писать ее.

Дэйв Ши и Молли Хольцшлаг, август 2004 г.

Благодарности

Дэйв Ши

Прежде всего, хочу сказать спасибо своей жене Эйприл за то, что она долгими вечерами мирилась с моим отсутствием, когда я работал над этой книгой в местной кофейне. Без ее поддержки, понимания и терпения я бы не справился.

(И, кстати, благодарю персонал кофейни Starbucks за то, что они так долго терпели парня, примостившегося в углу со своим ноутбуком, и поили его кофе с молоком.)

Конечно, я благодарен также своим родителям, Горду и Линде, и братьям Мэтту и Джереми за то, что они притворялись, будто понимают все написанное.

Выражаю признательность компании DreamFire Studios (www.dreamstudios.com) за то, что они предоставили хостинг для сайта Zen Garden и продолжают поддерживать его, хотя сайт сильно разросся.

Два года назад, когда я только начинал задумываться над проектом Zen Garden, я даже представить себе не мог, что вокруг него образуется такое сообщество. Друзья, вас слишком много, чтобы я мог поблагодарить каждого, но вы-то сами знаете о своих заслугах. То, что вы делаете ради процветания Web, вдохновляет и воодушевляет меня.

Огромное спасибо моему соавтору Молли за то, что она щедро делилась своими идеями и опытом и подбадривала меня. Ее перу принадлежит книга, которая привела меня в эту отрасль много лет назад; а совместная работа над этой книгой была истинным удовольствием.

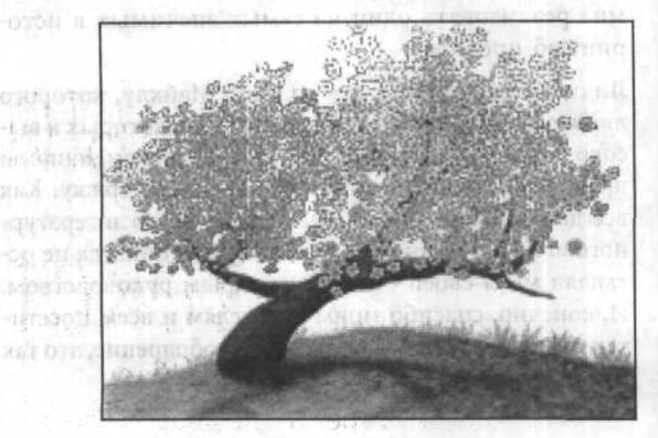
Молли Хольцшлаг

Удивительно работать в области, где наши студенты так быстро становятся нашими учителями, где наш труд может в мгновение ока изменить представление о Web. Именно таким опытом стало для меня совместное с Дэйвом Ши написание книги. Это дало мне шанс увидеть, как моя собственная работа вдохновила другого человека, который, в свою очередь, оказал влияние на целое новое поколение веб-дизайнеров. То, чему я научилась у Дэйва и многих других дизайнеров, внесших свою лепту в разработку сайта Zen Garden, позволило мне лучше понять, что такое CSS, более эффективно преподавать эту технологию и совершенно по-новому взглянуть на Web. Поэтому прежде всего я благодарна Дэйву, а затем уже всем прочим участникам проекта, многие из которых – мои друзья и учителя. Они помогли мне реализовать один из самых значимых в истории веб-проектов.

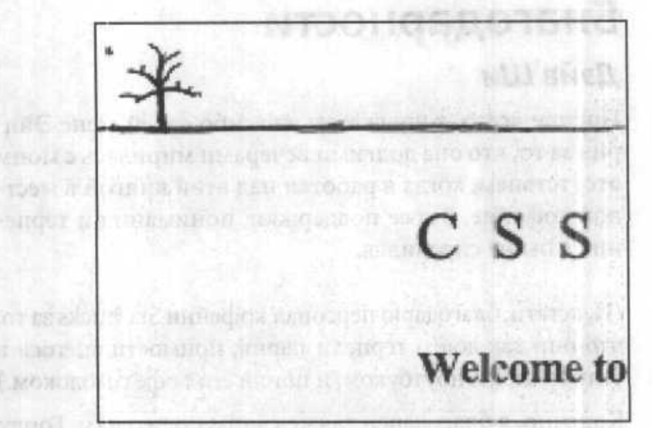
Лично же я хочу сказать спасибо Майклу, которого люблю, а также своей семье и друзьям, которых я высоко ценю и которые мирятся с моим переменчивым нравом: спасибо за вашу помощь и поддержку. Как всегда, благодарю Дэвида Фьюгейта, своего литературного агента и личного друга, который никогда не оставлял меня своей опекой и мудрым руководством. И, конечно, спасибо моим читателям и всем посетителям моего сайта – за поддержку и ободрение, это так много значит для меня.

Дэйв и Молли вместе

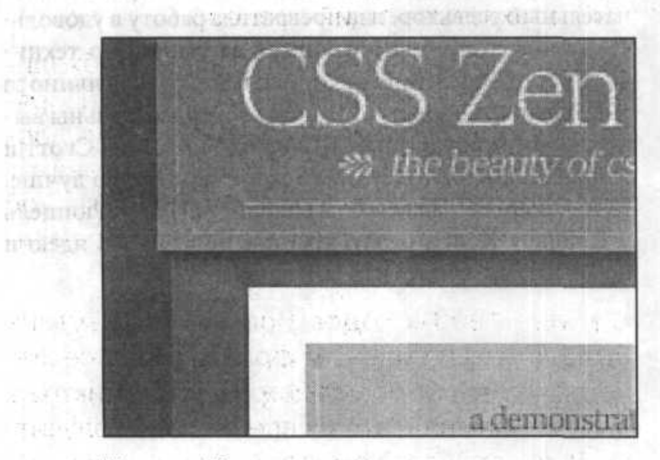
Всем принимавшим участие в подготовке этой книги огромное спасибо! Черил Ингланд – не просто замечательный редактор, она превратила работу в удовольствие! Благодарим Шон Инман за усилия по техническому оформлению рукописи и Элизу Рабеллино за тщательную корректуру. Мы также признательны выпускающим редакторам (Шарлин Уилл, Ким Скотт и Хилал Сала), благодаря им текст стал гораздо лучше. И, наконец, выражаем благодарность Нэнси Рюнцель и Клиффу Колби за то, что они поверили в идею и помогли нам воплотить ее в жизнь.



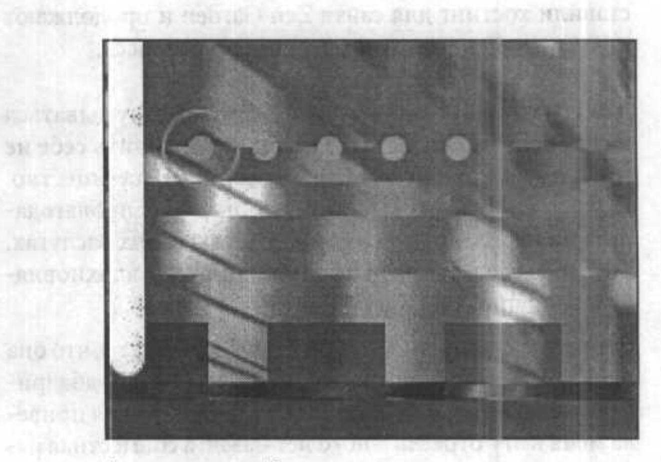
CSS Zen Garden, с. 14



Ветер, с. 18



Шторм, с. 19

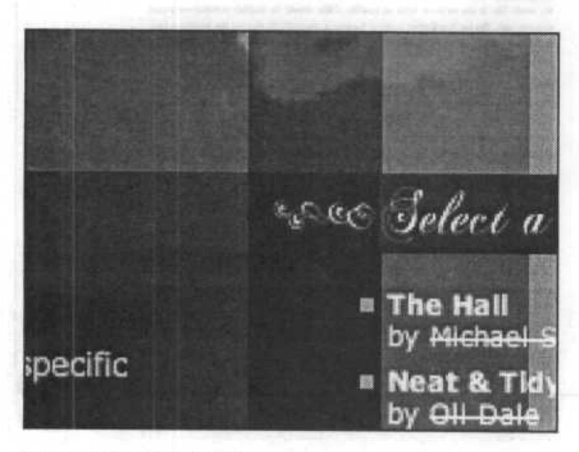


Четыре арки, с. 19



Исходный текст

Плавленый сыр с лососем, с. 19



Вариант дизайна, с. 50

За очень короткое время веб-дизайн продвинулся очень далеко. Дизайнерам буквально приходилось читать исходные тексты, действовать методом проб и ошибок и раздвигать пределы возможного. Это был спонтанный и захватывающий процесс.

Однако со временем стала ясна необходимость выработки стандартов Web. Это напрямую связано с самой важной для нас программой – веб-браузером. В этой главе вы познакомитесь со всем, что нужно знать об идеологии, процедурах и методах современного веб-дизайна.

В конце 1990-х годов Всемирная паутина была диким и опасным фронтиром. Все менялось, и все было возможно. Тогда никто из нас еще не знал, что из всего этого получится. И как это часто бывает с новой игрушкой, мы стремились «раздвинуть границы изведанного».

Возникновение

И мы их раздвигали. HTML возник как простой язык для разметки научных статей. Поставщики браузеров и графические дизайнеры, не найдя там средств для верстки, с презрением отвергли его. Для преодоления ограничений появились всякие ухищрения типа тега <FONT> и таблиц, все это шло вразрез с теми целями, для которых HTML первоначально разрабатывался.

Основные элементы применялись для создания макетов иначе, чем это задумывалось; элементы, имеющие очень конкретное назначение, игнорировались, так как выглядели невыразительно. На правильность структуры плевали в угоду визуальному украшению сайта.

Разумеется, в то время от основных производителей браузеров помощи было не дождаться. Казалось, что ситуация будет только ухудшаться. Два крупнейших игрока – Microsoft и Netscape – нагромождали одно нестандартное расширение на другое, стремясь обойти друг друга в знаменитой Войне Браузеров. Конкуренция, конечно, способствует инновациям, но может и задушить их на корню; веб-мастера были вынуждены поддерживать несколько версий сайта, поскольку одна и та же разметка редко выглядела одинаково в разных браузерах, если вообще хоть как-то отображалась.

В общем, был сплошной кавардак. Но огромные суммы, беспечно выброшенные на ветер во время бума интернет-компаний, все же способствовали росту Web, несмотря на имеющиеся проблемы. Обосновать расходы на поддержку двух версий сайта в условиях неограниченного финансирования было нетрудно. Однако на рубеже тысячелетия и авторы контента, и инвесторы решили, что с них хватит.

Начало перемен

В самом начале нового века бюджеты, выделяемые на поддержку сайтов, стали усыхать, поскольку капитал ушел. Когда в экономике начался спад, те немногие, кто сохранил работу, быстро поняли, что лишние усилия, затрачиваемые на написание кода для некорректно работающих и нестандартных браузеров, не окупаются. Нужно было что-то делать.

Была создана организация, объединившая широкие массы веб-дизайнеров и получившая название Web Standards Project (или WaSP, www.webstandards.org) (рис. 1). Она убеждала основных на тот момент производителей браузеров привести свои сильно отличающиеся программы в соответствие с «Рекомендациями», которые выпускал консорциум World Wide Web





The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. Send us a link to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

Requirements

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the Resources page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see submission guidelines), but we ask you release your CSS under a Creative Commons license identical to the one on this site so that others may learn from your work.

Bandwidth graciously donated by DreamFire Studios. Coming soon: Zen Garden, the book.

xhtml css cc 508 aaa

Ray Henry

FAQ

Translations

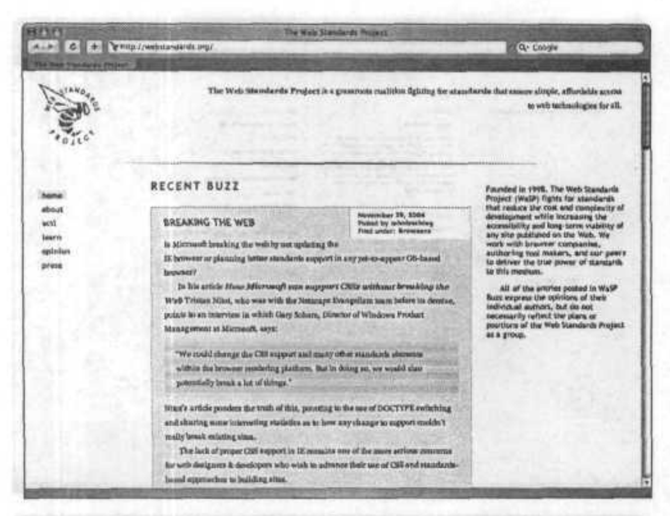


Рис. 1 ✦ Проект Web Standards Project

ПРИМЕЧАНИЕ

Консорциум W3C заинтересован в развитии Web. Занимаясь с 1996 года подготовкой рекомендаций, он стремится находить решения самых разнообразных технологических проблем, возникающих в Web.

Consortium (W3C). Стандартизация на основе единых спецификаций, разрабатываемых W3C, вселила в авторов веб-контента чувство некоторой стабильности. Со временем веб-дизайнеры уже могли создавать страницу со сложной HTML-разметкой, будучи уверены, что она будет выглядеть более-менее одинаково во всех браузерах и операционных системах.

Многие спецификации, выпущенные W3C, касаются архитектурных проблем и нетривиальных вопросов кодирования, с которыми среднему дизайнеру редко приходится сталкиваться. Тем не менее, начиная с первого стандарта HTML, принятого в середине 1990-х годов, W3C подготовил целый ряд спецификаций, стремясь решать проблемы совместимости по-умному, так, чтобы не ущемлять ни функциональности, ни доступности.

Зачем нужны стандарты?

В рабочие группы W3C входят специалисты из разных областей, которые собираются для того, чтобы заблаговременно решать проблемы, о которых средний дизайнер и думать никогда не должен. Цель состоит в выработке набора рекомендаций, настолько продуманных, что разработчикам редакторов и браузеров остается лишь строго следовать спецификации. В результате же авторы контента и дизайнеры обретают уверенность в том, что их сайты можно будет просмотреть с помощью широчайшего спектра пользовательских агентов.

Под «пользовательским агентом» мы, конечно, понимаем браузер, но браузер, установленный на типичном настольном компьютере, – это лишь верхушка айсберга. По мере распространения портативных устройств, таких, скажем, как мобильные телефоны, все больше пользователей станут просматривать веб-страницы, не будучи привязаны к своему домашнему компьютеру. Не все будут заходить в сеть с помощью вашего любимого браузера; просто потому, что не смогут. К примеру, для людей с физическими недостатками существуют специальные программы для чтения с экрана или печати азбукой Брайля; есть и программы для увеличения изображения.

Поскольку весь смысл выпуска спецификаций W3C состоял в том, чтобы навести порядок в неразберихе, царившей в Web в 90-х годах, теперь вряд ли имеет смысл готовить разные версии сайта для разных пользовательских агентов. Все это было учтено в различных рекомендациях, в частности HTML 4.01, XHTML, CSS (каскадных таблицах стилей) и DOM (объектной модели документа).

Эти технологии были с восторгом приняты дизайнерами и программистами, объединившимися в проекте Web Standards Project. Казалось логичным поддержать спецификации, которые гарантировали широчайшей аудитории доступность контента при сохранении контроля над визуальным представлением. Технология CSS попала в самую точку, но вместе с тем это был совершенно новый подход к созданию веб-сайтов. Не так-то просто оказалось убедить других дизайнеров в необходимости изучать CSS, и первые несколько лет нового тысячелетия ушли на то, чтобы понять, как этой методикой пользоваться.

Вот так обстояли дела в Web, когда зародилась идея сайта Zen Garden. В то время CSS был в основном уделом программистов и кодировщиков, они прекрасно разбирались в технических нюансах, но макеты страниц, выходившие из их рук, часто оказывались минималистскими, без искорки. Дизайнеры же в области графики не воспринимали идею CSS, поскольку не видели настоящих примеров CSS-дизайна.

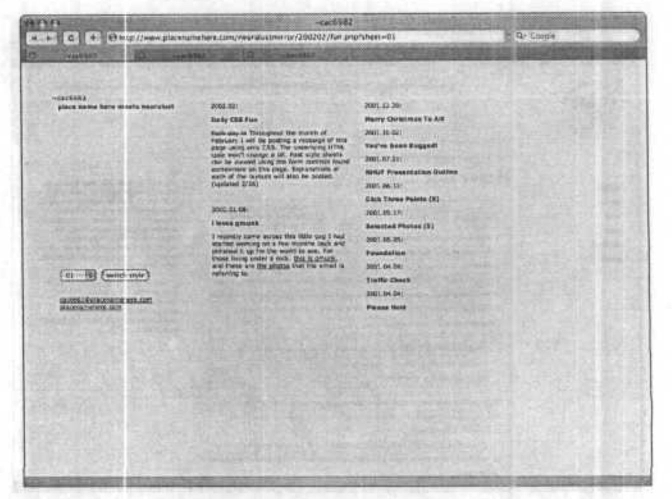
Засеваем почву

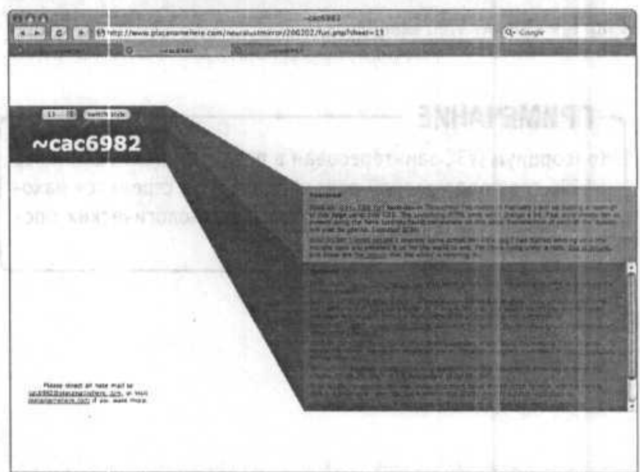
В конце 2002 года над этой проблемой задумался создатель сайта Zen Garden Дэйв Ши. Как далекие от изящества примеры могут вдохновить людей, способных создать с помощью CSS истинные шедевры? Поскольку у Дэйва был опыт и в кодировании, и в графике, то он сумел распознать потенциал CSS в качестве языка дизайнера. И понял, что им пользуются не те люди, которые должны бы.

В результате зародилась идея: а здорово было бы создать центральное хранилище лучших дизайнерских работ с применением CSS. Очевидно, недостаточно было просто собрать имеющееся, поскольку ничего *вообще* не было. Воодушевить дизайнеров на создание новых прекрасных работ – вот ключ к реализации идеи. Так были посеяны семена нового проекта.

Влияния со стороны

Один из основателей проекта Web Standards Project Джеффри Зельдман (Jeffrey Zeldman) в 2002 году написал статью, в которой заклинал людей, понимающих выгоду разработки на основе стандартов Web, не тратить время на убеждение других в достоинствах такого подхода, а просто начать применять его на практике. Именно так все преимущества можно продемонстрировать наиболее наглядно, поэтому фраза «Лучше один раз увидеть, чем сто раз услышать» стала





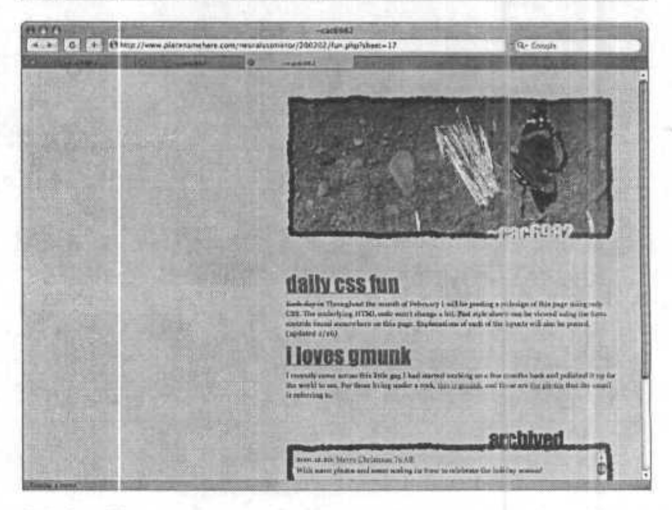


Рис. 2 ✦ Примеры того, что можно сделать с помощью таблиц стилей, с сайта Daily CSS Fun

ПРИМЕЧАНИЕ

Важно отметить, что стандарты Web – это не стандарты в традиционном понимании; никто не заставляет следовать им, и сам консорциум W3C скромно называет их рекомендациями. Однако на практике многие считают их настоящими стандартами, пусть даже они не носят такого названия. Подробнее об этом см. статью «What is a Web Standard» на странице www.webstandards.org/buzz/archive/2004_10.html#a000463.

ПРИМЕЧАНИЕ

Найти текст оригинальной статьи Зельдмана можно на странице www.zeldman.com/daily/0802c.html#Evangeline.

ПРИМЕЧАНИЕ

Проект Daily CSS Fun размещен на сайте www.placename-here.com/neuralustmirror/200202.

заповедью проекта. Зельдман же в дальнейшем написал авторитетное руководство по этой теме *Designing with Web Standards* (New Riders, 2003) («Web-дизайн по стандартам»).

В 2002 году Крис Качиано (Chris Cashiano) основал проект под названием Daily CSS Fun (Ежедневные CSS-приколы) (рис. 2). Идея заключалась в том, что на протяжении месяца Качиано ежедневно выкладывал новую таблицу стилей, которая весьма неожиданными способами представляла HTML-разметку. Он не называл себя графическим дизайнером, но результаты его работы убедительно показывали, сколь многого можно достичь путем применения для верстки одной лишь технологии CSS.

Вследствие неослабевающей популярности браузера Netscape Navigator 4, который плохо поддерживал CSS, все эти уроки оставались теорией – красивой, но далекой от практического применения в тогдашней Web. Ситуация начала изменяться по мере резкого сокращения присутствия Netscape Navigator 4 на рынке, к 2003 году он «исчез с экранов радаров». Настало время идти вперед.

Первые попытки

Идея сайта Zen Garden созрела, и в мае 2003 года Ши поместил плоды своих трудов на собственный сайт www.mezzoblue.com, а затем довольно быстро перевел их в домен www.csszengarden.com, где сайт существует и по сей день.

В конце 2002 года Ши уже начал разрабатывать предварительные прототипы Zen Garden, пользуясь четырьмя разными таблицами стилей, выражающими идею четырех античных стихий: земли, воды, огня и ветра. В то время разработка не была доведена до конца, и графика использовалась не слишком интенсивно. Как видно из рис. 3–6, первые наброски Zen Garden не особенно впечатляли.

Дизайн был скучным именно потому, что ему не хватало броской графики. Фоновые изображения были добавлены с помощью свойства `background-image`. Поместить же изображения на передний план не представлялось возможным. Применение тега `img` лишило бы дизайн гибкости, поскольку пришлось бы ограничиться единственным набором изображений. Одна лишь фоновая графика не решала проблемы, так как дизайнеру часто нужно, чтобы картинки перемежались с текстом, а не просто находились под ним. Поэтому фоновая графика давала лишь частичное решение.

Обходной путь был найден, когда Дуглас Боумен (Douglas Bowman) опубликовал ставшую теперь знаменитой статью «Использование фоновых изображений для замены текста». Тогда-то все и стало на свои места: поместить картинки на передний план можно было, привязав их к различным текстовым элементам на странице и заменив графическими эквивалентами. Технология CSS в совокупности с дополнительной настраиваемой разметкой позволяла заменить строчные текстовые элементы фоновыми изображениями – аналогично тому, как ведет себя тег `img` с подходящим атрибутом `alt`.

В апреле 2003 году началось серьезное планирование сайта Zen Garden. Был создан единый HTML-файл, к которому было применено пять разных вариантов дизайна (рис. 7–11).

ПРИМЕЧАНИЕ

Прочитав оригинальную статью Боумена на сайте Stopdesign (www.stopdesign.com/articles/replace_text), вы поймете, что у пропагандируемой в ней техники есть один недостаток – она плохо сочетается с доступностью. Но именно эта техника применяется во многих вариантах дизайна Zen Garden. Подробно мы рассмотрим ее в главе 4.



Рис. 3 ✦ Ранний прототип Zen Garden, земля

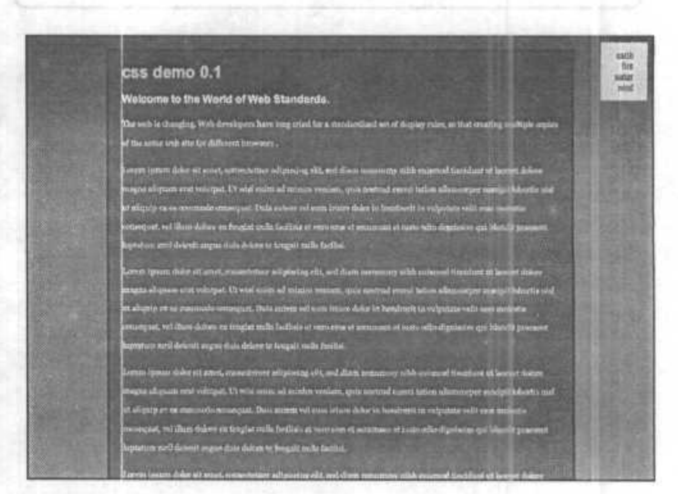


Рис. 4 ✦ Ранний прототип Zen Garden, огонь



Рис. 5 ✦ Ранний прототип Zen Garden, ветер

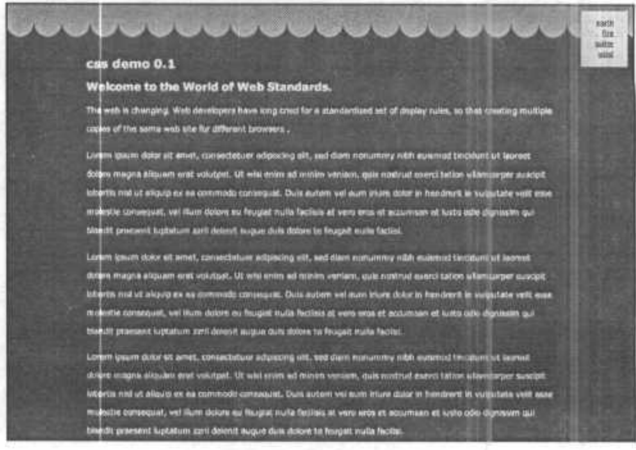


Рис. 6 ✦ Ранний прототип Zen Garden, вода

ПРИМЕЧАНИЕ

Поскольку изменить HTML-файл впоследствии уже было бы невозможно, авторы потратили немало времени на выверку разметки и текста, стремясь к тому, чтобы число элементов с уникальными значениями свойств `class` и `id` было достаточным для модификации с помощью таблиц стилей. После запуска сайта выявились некоторые проблемы с разметкой, которые нельзя было предвидеть заранее, – мы поговорим об этом ниже.

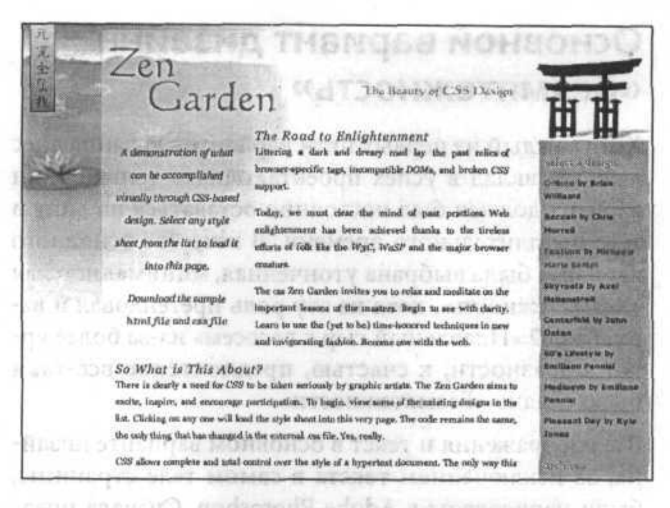


Рис. 7 ✦ #001, Безмятежность



Рис. 8 ✦ #002, Плавленый сыр с лососем



Рис. 9 ✦ #003, Шторм



Рис. 10 ✦ #004, Четыре арки

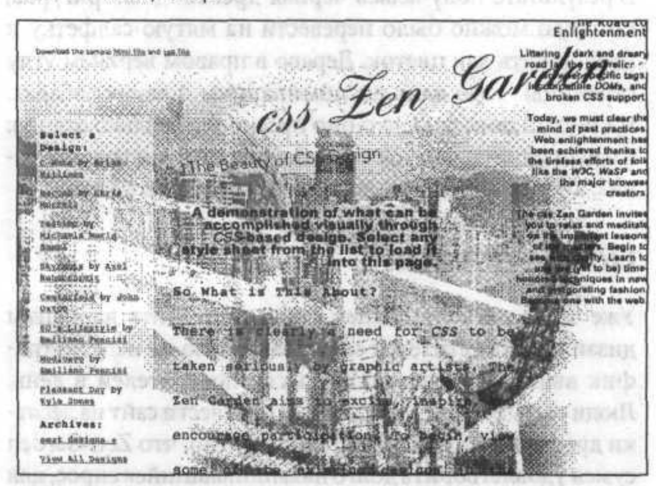


Рис. 11 ✦ #005, Жажда крови

Основной вариант дизайна – «Безмятежность»

Хотя каждый из первых пяти вариантов дизайна внес важный вклад в успех проекта, однако только один вариант должен был постоянно оставаться на виду в течение длительного времени. В качестве основного варианта была выбрана утонченная, минималистская «Безмятежность», хотя на эту роль претендовал и вариант 002 «Плавленый сыр с лососем» из-за более яркой образности; к счастью, предпочтение все-таки было отдано «Безмятежности».

Все изображения и текст в основном варианте дизайна, за исключением текста в самом теле страницы, были нарисованы в Adobe Photoshop. Сначала предполагалось, что тексты заголовков будут состоять из нарисованных вручную каллиграфических штрихов, но, потратив час на работу с кисточкой и тушью, Ши решил, что шрифт Calligraphic 421 от компании Bitstream подходит гораздо лучше (рис. 12).

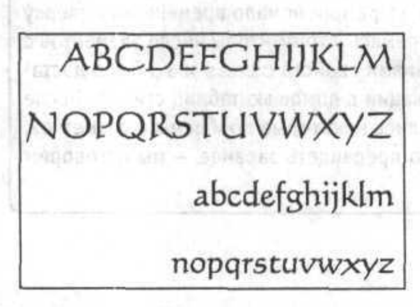


Рис. 12 ✦ Образцы шрифта Calligraphic 421

Китайским иероглифам присуща замысловатость и внутренняя красота, поэтому многие дизайнеры любят дополнять ими свой замысел. Пытаясь уйти от типичной западной привычки вставлять бессмысленный набор иероглифов для придания таинственности, Ши потратил некоторое время на то, чтобы иероглифы на вертикальном свитке, идущем вдоль левого края страницы, имели определенный смысл. И хотя идея была хороша, но результат все равно получился не слишком вразумительным: выбранные иероглифы означают «начало», «полный» или «целый» и «умение».

Помните, как детьми вы развлекались, капая тушью на лист белой бумаги и дуя на кляксу через соломинку? В результате получалась черная древовидная фигурка, которую можно было перевести на мятую салфетку и представить как цветок. Дерево в правом верхнем углу страницы – это дань воспоминаниям о таких забавах. На рис. 13 показано, что оно прошло несколько этапов. Сначала простыми цветами был нарисован грубый эскиз, а затем добавлено несколько слоев с деталями.

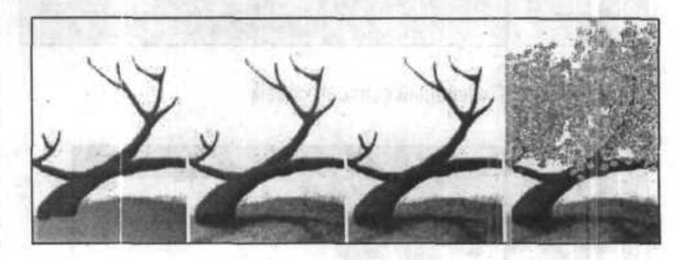


Рис. 13 ✦ Поэтапное рисование дерева

Последствия

Уже через несколько дней стали приходить варианты дизайна от посетителей сайта; за первую же неделю трафик вырос до нескольких тысяч посетителей в день. Люди вызывались добровольно перевести сайт на десятки других языков. Сразу стало понятно, что Zen Garden сумел удовлетворить долго накапливавшийся спрос, для которого в Web не было никаких других предложений.

С тех пор сайт Zen Garden упоминался в десятках журналов, издаваемых в разных странах, и во множестве книг, так что его канонический статус в истории веб-дизайна можно считать зафиксированным. Со всего света приходят благодарственные письма от дизайнеров и разработчиков, которым с помощью этого сайта удалось убедить своих работодателей, клиентов и коллег в ценности дизайна на основе CSS. А многие из тех, кто прислал свои работы, испытали резкое ускорение в своем карьерном росте, ощутив мощное воздействие сайта Zen Garden.

Хотя сайт Zen Garden создан и поддерживается одним человеком, он тем не менее является плодом коллективных усилий добровольцев. Своим успехом он обязан многим талантливым дизайнерам, приславшим свои работы. Каждый находит в Zen Garden что-то свое и, хотя этот сайт задумывался главным образом ради демонстрации ценности CSS для целей дизайна, но у него обнаружились и другие, заранее неведомые достоинства.

- Дизайнерам, решающим проблему верстки, теперь есть куда обратиться, если что-то не получается; когда в одном месте собраны сотни работ, шансы, что кто-то раньше уже сталкивался с подобной задачей, весьма высоки. Использовать представленные на сайте таблицы стилей в качестве образца для верстки допустимо и даже поощряется.
- Там же можно найти новые способы применения CSS и создания различных эффектов. Вполне вероятно, что среди сотен вариантов вы найдете для себя что-то новенькое.
- Это замечательный способ проверить поведение браузера. Весь код заведомо корректен и соответствует стандартам, так что теоретически страницы должны одинаково выглядеть в любом браузере. (На практике так обстоит дело не всегда. Из-за ошибок и отсутствия поддержки для некоторых элементов в конкретном браузере дизайнерам приходится учитывать возможные различия во внешнем виде, применяя для этого всяческие CSS-фильтры и уловки.)
- Именно сюда приводят работодателей и клиентов, когда хотят убедить их в пользе дизайна на основе стандартов.
- Это прекрасное лекарство от творческого застоя. Ничего не выходит? Загляните в поисках вдохновения.
- Это биржа труда как для высококвалифицированных профессионалов, так и для начинающих дизайнеров.

Работодатели и потенциальные клиенты регулярно обращаются к авторам понравившихся работ с предложением временного контракта или постоянной работы. Их произведения часто появляются на страницах книг и журналов.

✦ Преподаватели находят этот сайт полезным инструментом для обучения современным методам веб-дизайна.

Заложение фундамента

Еще до составления таблицы стилей необходимо определиться с разметкой своей веб-страницы. Раз вы читаете эту книгу, то, наверное, слышали об отделении структуры от представления. Для тех же, кому эта идея внове, скажем, что основное достоинство дизайна на основе CSS состоит в том, что таблицы стилей можно присоединить к имеющемуся хорошо структурированному документу, дабы сделать его визуально привлекательным.

Под форматированием веб-страниц долго понимали выбор нужных базовых элементов HTML в зависимости от того, как они *представляются.* Если вам нужен был абзац с отступом, следовало поместить текст внутрь тега `<blockquote>`, чтобы создать поля с обеих сторон. У элемента `<blockquote>`, однако, есть вполне конкретное назначение – подчеркнуть, что фрагмент текста является цитатой; отсюда и название. Применение этого элемента для выделения текста, не являющегося цитатой, просто ради создания отступа, означает неверный подход к задаче; ведь CSS позволяет сделать отступ для любого элемента.

ПРИМЕЧАНИЕ

Чтобы понимать, какой элемент использовать в конкретной ситуации, нужно знать основы языка HTML. В качестве практического руководства по разметке реальных документов мы рекомендуем книгу Дэна Сидерхольма «Решения на базе стандартов Web: справочник по разметке и стилям» (Web Standards Solution: Markup and Style Handbook. Friends of ED, 2004).

Правильное структурирование документа – это, по сути дела, выбор подходящих элементов, причем решение должно основываться на том, для чего элемент предназначен, а не на том, как он выглядит.

Теперь целью становится перенос всего форматирования в CSS-файл и использование HTML-разметки только для описания структуры. CSS – это язык дизайна, который накладывается на существующую структуру и позволяет визуализировать ее. Язык HTML более не следует использовать для дизайна.

Ну а все-таки – что мы выигрываем, уделяя такое внимание правильности структуры? Вообще, что в этом контексте означает слово «структура»?

Семантическая разметка

Разница между двумя идеологическими подходами: кодирование ради нужного представления и кодирование ради правильной семантики – может быть довольно тонкой, поэтому приведем пример. Следующий фрагмент можно назвать презентационной разметкой:

```
<br><br>
<b><font size="2">Наша семья</font></b>
<br><br>
<font size="1">На фотографии Мэтт и Джереми.
Как обычно, Мэтт корчит рожи. У нас не так
много фотографий, где бы он этого не делал.</
font>
```

А вот тот же текст, размеченный семантически:

```
<div id="family">
 <h3>Наша семья</h3>
 <p>На фотографии Мэтт и Джереми. Как обычно,
Мэтт корчит рожи. У нас не так много
фотографий, где бы он этого не делал.</p>
</div>
```

Грубо говоря, элемент – это часть структуры, тогда как тег – лишь синтаксическая конструкция, позволяющая описать структуру.

В первом примере все теги предназначены исключительно для форматирования. Теги
 нужны для вставки пустого места между строками, теги <b> – для отображения текста полужирным шрифтом, а теги <font> – для задания размера шрифта.

Во втором примере внешнему виду страницы не уделяется никакого внимания; упор сделан на то, насколько хорошо элементы h3 и p описывают назначение каждого фрагмента на странице. Элемент p содержит собственно текст, а элемент h3 описывает заголовок, который в контексте всей страницы имеет уровень 3.

Главное во втором примере – это то, что внешнее представление элементов на данном этапе несущественно; мы все равно переопределим его с помощью CSS. Элементы выбраны только исходя из соображений адекватного описания контента. Именно в этом и состоит семантика HTML – выбирать элементы, основываясь на их предназначении, а не внешнем виде.

У семантического форматирования документа есть весомые преимущества, которые мы обсудим чуть ниже. А пока имейте в виду, что применять таблицы стилей следует к правильному, хорошо семантически структурированному документу.

ПРИМЕЧАНИЕ

Какая разница между тегом <p> и элементом p? В некотором смысле они обозначают одно и то же, но есть тонкое различие. Тег – это конкретный разделитель в языке HTML; <p>, <div> и </body> – все это теги. Элемент состоит из открывающего и закрывающего тега, между которыми обычно находится некоторый контент; исключение составляет, например, тег
, одновременно являющийся открывающим и закрывающим, внутри него контента быть не может.

Создание подходящей разметки

Не существует универсальных правил создания документов, поскольку все страницы различны и должны структурироваться с учетом своих особенностей. Но некоторые общие соображения по HTML-разметке все же можно высказать.

Используйте DOCTYPE

Включив в начало разметки тег DOCTYPE (тип документа), вы сообщите браузеру или иному пользовательскому агенту, какой язык применяется для разметки документа. Каким бы языком вы ни пользовались, помещать в начало DOCTYPE – признак хорошего тона.

Пример:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
```

Задавайте язык и набор символов

Указание естественного языка документа не менее важно, чем задание DOCTYPE. Если вам приходилось встречать страницы с «крокозябрами», которые на самом деле написаны на японском, греческом, суахили или каком-то другом языке, то вы знаете, что происходит, когда автор документа забывает указать кодировку. Поскольку Google и другие поисковые машины фильтруют результаты по-разному в зависимости от языка запроса, то правильное указание кодировки окупается и тогда, когда вы хотите, чтобы поисковая машина возвращала ссылку на ваш сайт. Подробнее о кодировках символов мы поговорим в этой главе ниже.

Примеры:

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" >
```

Говорит, что данный документ составлен на языке с ISO-кодом en, то есть на английском.

```
<meta http-equiv="Content-Type" content="text/
html; charset=utf-8" />
```

Определяет кодировку символов для документа, в данном случае UTF-8.

ПРИМЕЧАНИЕ

Углубленное обсуждение вопроса об отделении структуры от представления см. в статье Роджера Йохансона (Roger Johansson) «Developing with Web Standards», опубликованной на сайте 456 Berea Street (www.456bereastreet.com/dwws).

Присваивайте заголовки

На каждой странице должен быть элемент <title> с содержательным заголовком. Заголовки служат для описания ссылки, на которую в браузере установлена закладка, а поисковые машины повышают приоритет ключевых слов, встречающихся в заголовке, поэтому во многих случаях документ оказывается выше в списке найденных. По возможности каждой странице сайта следует приписывать уникальный заголовок, который описывает именно данную страницу, а не сайт в целом.

Пример:

```
<title>css Zen Garden: The Beauty in CSS
Design</title>
```

Пользуйтесь подходящими элементами

Мы уже касались этой темы, но нелишне будет повторить: используйте для разметки такие элементы HTML, которые описывают структуру документа, а не его внешний вид. Элемент p служит для выделения абзацев, а не для разбиения строки. Элемент blockquote нужен для оформления цитат, а не для формирования отступа. И так далее.

Конечно, не все элементы HTML вошли в схему XHTML 1.0 Strict. В ней (а равно и в схеме XHTML 1.1) меньше элементов, зато они служат более четко выраженным целям. Если не существует элемента, отражающего назначение структурной единицы в вашем документе, то стоит вспомнить об универсальных элементах div или span. Но тут есть ловушка... впрочем, читайте дальше.

Пример:

```
<h3>The Road to Enlightenment</h3>
```

а не:

```
<code style="font-size: 1.5em">The Road to
Enlightenment</code>
```

> **ПРИМЕЧАНИЕ**
>
> Полный список элементов, включенных в схему XHTML 1.0, можно найти в разделе «XHTML 1.0 Reference» на сайте W3Schools (http://w3schools.com/xhtml/xhtml_reference.asp).

Избегайте чрезмерного употребления <div>

При использовании элементов div и span есть опасность переборщить. В небольшом количестве и размещенные в стратегически важных местах, они способны существенно улучшить структуру документа.

Если же их оказывается чересчур много, то стоит спросить себя, нельзя ли решить задачу с помощью более подходящих HTML-элементов. Например, никогда не следует пользоваться элементом `div` там, где более уместен `h3`, а `span` не рекомендуется употреблять вместо `label`.

Но это вовсе не значит, что от этих элементов надо отказаться. Несколько `div` в подходящих местах документа позволяют лучше контролировать стили и логически разграничить разделы. Считайте `div` повторно используемым контейнером: рассовывать контент по слишком многим контейнерам неудобно, зато несколько удачно размещенных контейнеров помогут рассортировать контент так, как вам хочется.

Пример:

```
<div id="pageHeader">
 <h1>css Zen Garden</h1>
 <h2>The beauty of<acronym title="Cascading
Style Sheets">CSS</
 acronym> Design</h2>
</div>
```

Меньше разметки

В предыдущих двух советах неявно присутствует мысль о том, что чем меньше разметки, тем лучше. Так оно и есть. Если вы хотите выстроить солидную конструкцию, то не надо загромождать ее разметкой. Используйте лишь те элементы, что необходимы, а все прочее отбросьте. Это позволит уменьшить размер файла (а, стало быть, и время загрузки), кроме того, браузеру будет проще интерпретировать файл и, значит, он быстрее появится на экране.

Пример:

```
<p>The Zen Garden aims to excite, inspire, and
encourage participation.</p>
```

а не:

```
<div>
 <p><span class="text">The Zen Garden aims to
excite, inspire, and encourage
participation.</span></p>
</div>
```

Правильно пользуйтесь атрибутами class и id

Если добавить к элементам идентифицирующие атрибуты, то позднее на них можно будет сослаться из таблицы стилей или из сценария на языке JavaScript. Атрибут `class` с одним и тем же значением может встречаться у разных элементов на странице, тогда как каждое значение атрибута `id` в пределах страницы должно быть уникальным.

У одного элемента может быть несколько классов, а один класс может применяться к нескольким элементам. Можно применить к элементу одновременно атрибуты `class` и `id`, но не забывайте, что наличие на странице более одного элемента с одним и тем же значением `id` не допускается. Имена классов и значения идентификаторов могут состоять из букв и цифр (a–z, A–Z, 0–9), но не должны начинаться с цифры. И лучше считать, что те и другие чувствительны к регистру, хотя некоторые браузеры не обращают на это внимания.

Три примера правильных идентификаторов:

```
<body class="homepage">
<p id="introduction">
<div id="section5" class="top corner solid">
```

Два примера неправильных идентификаторов:

```
<span class="15pxhigh">
<div id="footer"><div id="footer">
```

Испытание временем

Если вам доводилось писать документ, рассчитанный на сколько-нибудь продолжительное существование, то вы понимаете, как важно внимательно подойти к его составлению.

Рассмотрим, например, издательское дело. Эта книга выдержала много правок и рецензий, нацеленных на то, чтобы проверить все факты и улучшить стиль. Поскольку книга обычно стоит на книжной полке куда дольше, чем веб-страница находится в кэше браузера, то всю эту работу надо проделать заранее, чтобы рукопись оказалась безупречной. (Постучим по дереву!)

В Web таких ограничений нет, и это создает как удобства, так и трудности. Сервер возвращает веб-страницу при каждом обращении со стороны посетителя. Обычно это означает, что вы можете вносить изменения в дизайн уже после публикации. При последующем посещении сайта можно увидеть эти изменения, книга же останется неизменной вплоть до переиздания. Искушение продолжить совершенствование сайта зачастую очень сильно, к досаде дизайнеров, подписавших контракт с фиксированным сроком действия.

Хотя сайт Zen Garden и существует в Web, но у него есть особенности, из-за которых составление HTML для него оказалось ближе к написанию книги. Если бы было разрешено изменять разметку после того, как посетители начали присылать свои варианты, то все старые поступления оказались бы неработоспособны,

и сайт перестал бы отвечать своему назначению. В результате модификации текста изменился бы размер документа, а макет страницы мог бы непредсказуемо «разъехаться». Любые структурные или визуальные изменения были бы пагубны, поэтому HTML должен оставаться таким, как в момент запуска сайта.

Исходный HTML-текст сайта Zen Garden

Считайте, что HTML-структура сайта Zen Garden так же неизменна, как текст книги. Она не изменится никогда, хотя улучшить ее можно было бы миллионом способов. Сам контент может быть слегка модифицирован, но основные HTML-элементы «высечены в камне».

На следующих нескольких страницах приведен исходный HTML-текст Zen Garden, на который мы будем ссылаться в разных местах[1].

Получить электронную копию можно также, зайдя на сайт www.csszengarden.com и выбрав из меню пункт **View Source** (Текст HTML). Во всех основных браузерах она расположена примерно в одном месте: в меню **View** (Вид). Для простоты можно вместо этого найти на самом сайте Zen Garden ссылку на образец HTML-файла.

Из листинга изъяты комментарии, имеющиеся в оригинальном исходном тексте. Вместо них на полях приведены примечания к некоторым разделам, поясняющие, почему было сделано так, а не иначе. Если вы хотите увидеть оригинальные комментарии, зайдите на сайт.

```
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0
Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" >
<head>

<meta http-equiv="content-type" content="text/
html;
charset=iso-8859-1" />
<meta name="author" content="Dave Shea" />
<meta name="keywords" content="design, css,
cascading, style, sheets, xhtml, graphic
design, w3c, web standards, visual, display" />
<meta name="description" content=
"A demonstration of what can be accomplished
visually through CSS-based design." />
<meta name="robots" content="all" />

<title>css Zen Garden: Красота CSS-дизайна</
title>
```

Элемент DOCTYPE необходим для описания типа документа. См. раздел «Используйте DOCTYPE» выше в этой главе.

Задание примененной в документе кодировки символов исключительно важно в глобальной среде, каковой является Всемирная паутина. Подробнее см. раздел «Кодировки символов» ниже в этой главе, а более широкое обсуждение этой темы можно найти в статье Джоэля Спольски (Joel Spolsky) «The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)» на сайте www.joelonsoftware.com/articles/Unicode.html.

[1] Для удобства читателя HTML-текст переведен на русский язык, хотя на всех снимках экрана представлен оригинальный английский вариант. – *Прим. пер.*

Этот пустой элемент script нужен для того, чтобы обойти одну ошибку в Microsoft Internet Explorer для Windows. См. статью «Flash of Unstyled Content» на сайте Blue Robot (www.bluerobot.com/web/css/fouc.asp).

```
<script type="text/javascript"></script>
```

Этот метод импорта таблицы стилей уже не является необходимым, раньше он гарантировал, что Netscape Navigator 4 не будет пытаться применить внешний CSS-файл вместо отображения простой нестилизованной разметки. См. статью Эрика Мейера (Eric Meyer) «Tricking Browsers and Hiding Styles» на странице www.ericmeyeroncss.com/bonus/trick-hide.html.

```
<style type="text/css" media="all">
 @import "sample.css";
</style>
</head>
```

Идентификатор (атрибут id) в элементе body называется CSS-сигнатурой. Пользовательская таблица стилей может переопределить все или некоторые стили на странице в зависимости от указанной сигнатуры. См. статью «[css-d] CSS signatures» (http://archivist.incutio.com/viewlist/css-discuss/13291).

```
<body id="css-zen-garden">
```

Поместив логически связанные группы элементов в отдельные div, мы упростим и сделаем более гибким процесс верстки.

```
<div id="container">
  <div id="intro">
    <div id="pageHeader">
      <h1><span>css Zen Garden</span></h1>
      <h2><span>Красота <acronym
title="Каскадные таблицы стилей">
      CSS</acronym>-дизайна</span></h2>
    </div>

    <div id="quickSummary">
      <p class="p1"><span>Демонстрация того,
чего можно достичь в плане визуального
представления в дизайне на основе <acronym
      title="Каскадные таблицы стилей">CSS</
acronym>. Выберите любую таблицу стилей из
списка и посмотрите, как будет выглядеть
страница.</span></p>
```

Всегда стремитесь к простоте. Хотя люди, желающие внести свой вклад в Zen Garden, наверняка понимают, как загрузить файлы с примерами, имея эти ссылки (щелкнуть правой кнопкой мыши и выбрать из контекстного меню пункт **Save As** (Сохранить как)), но те, кто просто проявляет интерес к коду, могут этого и не знать. По электронной почте продолжают приходить письма с вопросами, как сохранить эти файлы. Это означает лишь, что ваша аудитория состоит не только из тех, на кого вы рассчитывали.

```
        <p class="p2"><span>Загрузите образец
<a href="zengarden-sample.html"
title="Исходный HTML-код этой страницы, не
модифицировать.">HTML-файла</a> и
<a href="zengarden-sample.css"
      title="Образец CSS для этой страницы,
этот файл можно модифицировать.">CSS-файла</
a></span></p>
    </div>
```

Наверное, было бы разумнее закончить здесь элемент #intro div и поместить #preamble внутрь #supportingText. Длина раздела #preamble наводит на мысль о том, что его место в более крупном блоке текста, составляющем основное содержание страницы, а не во введении. С точки зрения позиционирования и применения стилей это тоже обеспечило бы дополнительную гибкость.

```
    <div id="preamble">
      <h3><span>Путь к прозрению</span></h3>
      <p class="p1"><span>Обочина темной и
унылой дороги усыпана останками зависимых от
браузеров тегов, несовместимых моделей
      <acronym title="Document Object
Model">DOM</acronym> и некорректно
реализованной поддержки <acronym
title="Cascading Style Sheets">CSS</
acronym>.</span></p>
```

```
    <p class="p2"><span>Ныне мы должны
очистить свой ум от прошлого опыта. Прозрение
снизошло на нас благодаря неустанным усилиям
людей из <acronym title="World Wide Web
Consortium»>W3C
    </acronym>, <acronym title="Web Standards
Project">WaSP</acronym> и создателей основных
браузеров.</span</p>

    <p class="p3"><span>css Zen Garden
приглашает вас отдохнуть душой и поразмышлять
над важными уроками мастеров. Пусть снизойдет
на вас просветление. Научитесь пользоваться
проверенными временем (впрочем, это еще
впереди) методами по-новому. Станьте частью
Всемирной паутины.</span</p>
   </div>

  </div>

  <div id="supportingText">
   <div id="explanation">
    <h3><span>О чем этот сайт?</span></h3>
    <p class="p1">Существует очевидная
необходимость в том, чтобы художники-графики
всерьез восприняли идею <acronym title=
    "Cascading Style Sheets">CSS</acronym>.
Цель сайта Zen Garden - пробудить в них
энтузиазм и призвать к участию. Для начала
ознакомьтесь с уже существующими вариантами
дизайна, которые перечислены в списке. Выбор
любого варианта приведет к загрузке этой же
страницы, но с другой таблицей стилей.
Разметка остается той же самой, меняется лишь
внешний .css-файл. Честное слово.
    </span></p>


    <p class="p2"><acronym title="Cascading
Style Sheets">CSS</acronym> обеспечивает
полный контроль над стилем гипертекстового
документа. Единственный способ доказать
истинность этого утверждения, так, чтобы у
зрителя осталось глубокое впечатление,
показать, чего можно достичь, когда
инструментом пользуется человек, способный
превратить голую структуру в произведение
искусства. На сегодня большинство трюков
демонстрируют "структурщики" и кодировщики.
Дизайнерам еще только предстоит внести свою
лепту. Такое положение дел должно быть
изменено.</span></p>
   </div>


   <div id="participation">
    <h3><span>Приглашение к участию</span></h3>
    <p class="p1"><span>Рады приветствовать
всех художников-графиков. Раз вы собираетесь
модифицировать эту страницу, необходимо
глубоко разбираться в технологии <acronym
title="Cascading Style Sheets">CSS</acronym>,
но файлы с примерами достаточно хорошо
прокомментированы, так что даже начинающие
пользователи <acronym title="Cascading Style
```

Педанты могут возразить, что W3C – это аббревиатура, а не акроним. Это действительно так, и в HTML есть элемент abbr для такого рода сокращений, но Internet Explorer его не поддерживает, так что многие пользуются элементом acronym, который поддержан повсеместно.

Нет, мы не знаем, что означает слово «структурщик». Когда писался этот текст, существовала настоятельная потребность провести четкое различие между теми, кто пишет код, и теми, кто размечает HTML-документы. Но почему такая потребность возникла, мы уже не помним.

В английском оригинале встречается апостроф, который представлен в виде последовательности ’. Это код символа для настоящей типографской одиночной кавычки (') вместо прямого апострофа ('), который обычно используется в качестве замены. В Web поддерживается множество типографских символов и соглашений, этим надо пользоваться при малейшей возможности. См. статью «The Trouble with EM'n EN (and Other Shady Characters)» на сайте www.alistapart.com/articles/emen.

Требования к присылаемым работам изменились вследствие непредвиденно большого числа предложений. Прежде чем отправлять свое произведение, ознакомьтесь с «Инструкцией для авторов» на странице www.mezzoblue.com/ zengarden/submit/guidelines.

```
Sheets">CSS</acronym> смогут взять их за
отправную точку. Посетите <a href="http://
www.mezzoblue.com/zengarden/resources/" title=
"Список относящихся к CSS ресурсов»>
Руководство по ресурсам, посвященным <acronym
title="Cascading Style Sheets">CSS</acronym></
a>, где вы найдете ссылки на подробные
руководства и сборники советов о том, как
работать с <acronym title="Cascading Style
Sheets">CSS</acronym>.</span></p>
<p class="p2"><span>Вы можете модифицировать
таблицу стилей как угодно, но не трогайте
<acronym title="HyperText Markup
Language">HTML</acronym>-разметку. This may
seem daunting at first if you&#8217;ve never
worked this way before, but follow the listed
links to learn more, and use the sample files
as a guide'.</span></p>

   <p class="p3"><span> Загрузите образец <a
href="zengarden-sample.html" title="Исходный
HTML-код этой страницы, не
модифицировать.">HTML-файла</a> и <a
href="zengarden-sample.css" title="Образец CSS
для этой страницы, этот файл можно
модифицировать.">CSS-файла</a>, чтобы работать
локально. Закончив свой шедевр (пожалуйста, не
присылайте незавершенные работы), разместите
.css-файл на веб-сервере, который вы можете
контролировать. <a href="http://
www.mezzoblue.com/zengarden/submit/" title="C
помощью этой формы можно сообщить нам о своем
файле">Пришлите нам ссылку</a> на свой файл и,
если мы решим выставить его на всеобщее
обозрение, то робот скачает связанные с ним
изображения. В конце концов результат будет
размещен на нашем сервере.</span></p>
  </div>

  <div id="benefits">
    <h3><span>Выгоды</span></h3>
    <p class="p1"><span>Зачем нужно
становиться участником проекта? Чтобы о вас
узнали, чтобы припасть к источнику вдохновения
и чтобы было на что сослаться, когда речь
зайдет о дизайне на основе <acronym
title="Cascading Style Sheets">CSS</acronym>.
А это ой как необходимо, даже и сегодня. Все
больше крупных сайтов переходят на эту
технологию, но их еще мало, мало... Когда-
нибудь наша галерея станет историческим
курьезом, но пока этот день еще не настал.</
span></p>
  </div>

  <div id="requirements">
    <h3><span>Требования</span></h3>
    <p class="p1"><span>Мы хотели бы, чтобы
```

[1] В тексте книги переведены все листинги сайта, однако здесь они оставлены в оригинале, чтобы проще было понять работу с апострофами. – *Прим. пер.*

```
таблицы стилей по возможности соответствовали
спецификации <acronym title="Cascading Style
Sheets, version 1">CSS1</acronym>. Применение
<acronym title="Cascading Style Sheets,
version 2">CSS2</acronym> следует ограничить
только элементами, которые широко поддержаны.
Сайт css Zen Garden призван продемонстрировать
практически применимые приемы работы с
<acronym title="Cascading Style Sheets">CSS</
acronym>, а не трюки, которые сможет оценить
лишь 2% аудитории. Единственное настоящее
требование, которое мы предъявляем, - это
корректность вашей таблицы стилей.</span></p>

    <p class="p2"><span>К сожалению, при
таком подходе к дизайну выявляются ошибки в
различных реализациях <acronym title=
"Cascading Style Sheets, version 1">CSS1</
acronym>. Иногда разные браузеры отображают
одну и ту же страницу по-разному, несмотря на
то что таблица стилей абсолютно корректна.
Особенно досадно, когда после исправления
одной ошибки перестает работать что-то другое.
На странице <a href="http://www.mezzoblue.com/
zengarden/resources" title="Список относящихся
к CSS ресурсов">Ресурсы</a> есть информация о
некоторых ошибках и способах их обхода. Полное
соответствие браузера спецификации все еще
остается несбыточной мечтой, и мы не ожидаем,
что ваш код будет отображаться на всех
платформах одинаково с точностью до пикселя.
Но если ваш дизайн не отображается нормально,
по крайней мере в браузерах IE5+/Win и
Mozilla (ими пользуется свыше 90% населения),
то, скорее всего, мы его не примем.</span></p>
    <p class="p3"><span>Мы просим присылать
только оригинальные работы. Уважайте законы об
авторском праве. Старайтесь свести к минимуму
объем спорных материалов - показанная со
вкусом обнаженная натура приемлема,откровенная
порнография будет отвергнута.</span></p>

    <p class="p4"><span>Этот сайт - в равной
мере выставка и учебный класс. Вы сохраняете
все авторские права на свою графику (с
немногими исключениями, перечисленными в<a
href="www.mezzoblue.com/zengarden/submit/
guidelines/">инструкции авторам</a>), но
публиковать свою таблицу стилей мы просим на
условиях лицензии <a href="http://
creativecommons.org/licences/sa/1.0/"
title="Лицензия, действующая на сайте Zen
Garden">Creative Commons</a>, чтобы другие
люди могли чему-то научиться у вас.</span></p>
    <p class="p5"><span>Хостинг любезно
предоставлен компанией <a href="http://
www.dreamfirestudios.com/">DreamFire Studios
    </a>. Купите <a href="/book/">Книгу</
a>!</span></p>
</div>
```

В главе 7 вы увидите, как можно поэкспериментировать с продвинутыми CSS-эффектами, «подсунув» в то же время старым браузерам урезанную таблицу стилей. В некоторых вариантах дизайна эта идея нашла воплощение, так что требования к поддержке со стороны браузера можно было бы сформулировать так: «На ваше усмотрение».

Хотя присланные работы часто проверяются в разных браузерах перед размещением на сайте Zen Garden, все же тестирование входит в обязанности автора. Если вы заметите погрешности в макете страницы, помните, что работы приходят со всего света. Иногда у автора просто нет доступа к тем браузерам и операционным системам, с которыми работаете вы.

Перечень ссылок в разделе `#footer` – это список... так почему бы не воспользоваться для его разметки элементом `ul`? Риторический вопрос – надо было бы оформить их в виде какого-то списка.

```
<div id="footer">
    <a href="http://validator.w3.org/check/
referer" title="Проверить корректность XHTML-
разметки на этом сайте">xhtml</a>  
    <a href="http://jigsaw.w3.org/css-
validator/check/referer" title="Проверить
корректность CSS на этом сайте">CSS</a>  
    <a href="http://creativecommons.org/
licences/by-nc-sa/1.0/" title="Ознакомиться с
лицензией для этого сайта, с разрешения
Creative Commons.">cc</a>  
    <a href="http://bobby.watchfire.com/
bobby/bobbyServlet?URL=http%3A%2F%2Fwww.
mezzoblue.com%2Fzengarden%2F&amp;output=
Submit&amp;gl=sec508&amp;test="
title="Проверить доступность этого сайта
согласно U.S. Section 508">508</a>  
    <a href="http://bobby.watchfire.com/
bobby/bobbyServlet?URL=http%3A%2F%2Fwww.
mezzoblue.com%2Fzengarden%2F&amp;output=Submit
&amp;gl=wcag-aaa&amp;test=" title="Проверить
доступность этого сайта согласно Рекомендациям
по доступности контента WAI">aaa</a>
 </div>

</div>


<div id="linkList">
 <div id="linkList2">

    <div id="lselect">
     <h3 class="select"><span>Выберите
дизайн:</span></h3>
     <ul>
       <li><a href="/" title="AccessKey: a"
accesskey="a">Пример 1</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
```

В первоначальной разметке Zen Garden вместо элементов `li`, окружающих каждую ссылку, использовались `span`, но автоматизированные службы проверки доступности типа Bobby (http://bobby.watchfire.com) сочли это ошибкой. Дополнительные пробелы ` ` в конце каждой ссылки – это следы неудачной попытки обойти проблему; очень скоро `span` были заменены списком, но уже ненужные ` ` так и остались.

```
<li><a href="/" title="AccessKey: b"
accesskey="b">Пример 2</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
<li><a href="/" title="AccessKey: c"
accesskey="c">Пример 3</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
<li><a href="/" title="AccessKey: d"
accesskey="d">Пример 4</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
<li><a href="/" title="AccessKey: e"
accesskey="e">Пример 5</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
<li><a href="/" title="AccessKey: f"
accesskey="f">Пример 6</a>, автор
```

```
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
<li><a href="/" title="AccessKey: g"
accesskey="g">Пример 7</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
<li><a href="/" title="AccessKey: h"
accesskey="h">Пример 8</a>, автор
<a href="http://www.mezzoblue.com/"
class="c">Дэйв Ши</a> </li>
    </ul>
   </div>


   <div id="larchives">
    <h3 class="archives"><span>Архивы:</
span></h3>
    <ul>
     <li><a href="/" title="Следующий набор
вариантов дизайна. AccessKey: n"
accesskey="n"><span class="accesskey">n
</span>ext designs &raquo;</a> </li>

     <li><a href="/" title="Предыдущий набор
вариантов дизайна.
     AccessKey: p" accesskey="p">&laquo;
<span class="accesskey">p</span>previous
designs</a></li>
     <li><a href="http://www.mezzoblue.com/
zengarden/alldesigns" title="Все варианты
дизайна Zen Garden. AccessKey: w"
accesskey="w">Vie<span class="accesskey">w</
span> All
     designs</a></li>
    </ul>
   </div>

   <div id="lresources">
    <h3 class="resources"><span>Ресурсы:</
span></h3>
    <ul>


     <li><a href="www.csszengarden.com/001/
001.css" title= "Посмотреть исходный CSS-файл
для текущего варианта дизайна, AccessKey: v"
accesskey="v"><span class=
"accesskey">V</span>iew this
design&#8217;s<acronym title= "Cascading Style
Sheets">CSS</acronym></a></li>

     <li><a href="www.mezzoblue.com/
zengarden/resources" title= "Ссылки на сайты,
посвященные работе с CSS. AccessKey: r"
accesskey="r"><acronym title="Cascading Style
Sheets">CSS
     </acronym><span class="accesskey">R</
span>esources</a></li>
     <li><a href="www.mezzoblue.com/
zengarden/faq" title= "Перечень часто
задаваемых вопросов о Zen Garden. AccessKey: q"
accesskey="q"><acronym title="Frequently
Asked Questions">FA<span class="accesskey">Q</
span>
```

Клавиши доступа (accesskey) упрощают навигацию по сайту, но этой технологии присущи некоторые недостатки. См. статью «Using Accesskeys – Is it worth it?» (Стоит ли пользоваться клавишами доступа?) на странице www.wats.ca/articles/accesskeys/19.

Теперь, когда накопилось больше 100 вариантов дизайна, такой метод перебора выглядит смешным и нелепым. Гораздо полезнее ссылка на весь список вариантов (www.mezzoblue.com/zengarden/alldesigns). Но часто не замечают даже того, что есть. Многие полагают, что у сайта Zen Garden всего восемь вариантов дизайна, и не заглядывают в архивы. Не повторяйте эту ошибку!

Хотя пункт меню **View Source** известен всем, кто сколько-нибудь долго работал в Web, но новички часто присылают письма с просьбой сказать, как просмотреть CSS-файл для конкретного дизайна. Поэтому вскоре после запуска сайта я добавил эту ссылку.

В некоторых вариантах дизайна эти пустые элементы div и span служат для добавления графических изображений, которые почему-либо неудобно присоединять к остальным элементам. Считайте, что это повторно используемые элементы img.

```
          </acronym></a></li>
          <li><a href="www.mezzoblue.com/
zengarden/submit" title= "Присылайте свой CSS-
файл. AccessKey: s" accesskey="q">
          <span class="accesskey">S</span>ubmit a
Design</a></li>
          <li><a href="www.mezzoblue.com/
zengarden/translations/" title="Посмотреть
версии этой страницы на других языках.
AccessKey: t" accesskey="t"><span
class="accesskey">T
          </span>ranslations</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>

<div id="extraDiv1"><span></span></div>
<div id="extraDiv2"><span></span></div>
<div id="extraDiv3"><span></span></div>
<div id="extraDiv4"><span></span></div>
<div id="extraDiv5"><span></span></div>
<div id="extraDiv6"><span></span></div>
</body>
</html>
```

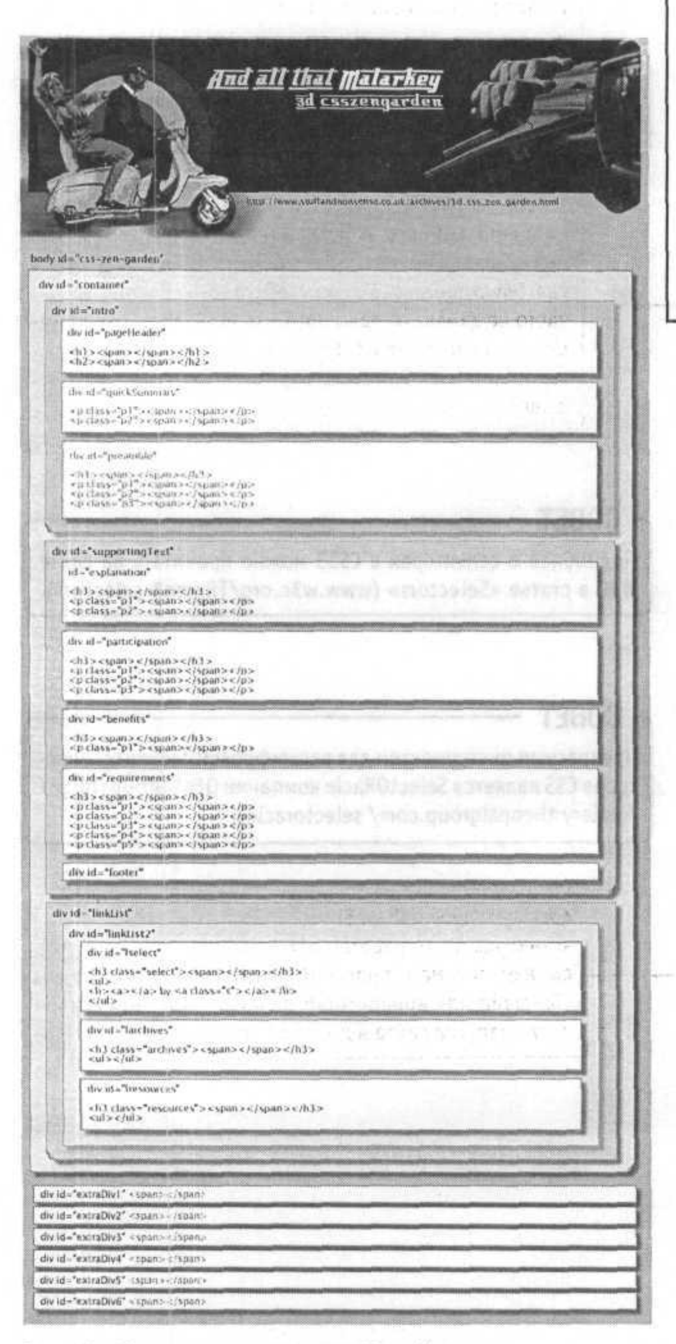


Рис. 14 ✦ Визуальное руководство Энди Кларка по структуре разметки сайта Zen Garden

Визуальная структура

Дизайнер Энди Кларк (Andy Clarke) создал визуальное руководство по структуре разметки сайта Zen Garden (рис. 14). На рисунке прекрасно видно, как различные элементы вкладываются друг в друга; он очень полезен в качестве отправной точки для изучения разметки сайта. См. также сопроводительную статью Кларка «3D CSS Zen Garden» (Трехмерная визуализация сайта CSS Zen Garden) (www.stuffandnonsense.co.uk/archives/3d_css_zen_garden.html).

Гибкость дизайна

С самого начала одной из основных целей при составлении разметки было обеспечение гибкости для создания различных вариантов дизайна. Необходимо было вставить в разметку достаточно много адресуемых элементов, чтобы внешний вид каждого можно было при необходимости модифицировать. Без дополнительного кода выбрать отдельный элемент на странице было бы невозможно. Селекторы потомков допускают следующий общий способ адресации элементов:

```
#quickSummary p {
  color: red;
}
```

Но при этом все элементы p внутри раздела #quickSummary становятся красными. А если нужно модифицировать только второй абзац внутри #quickSummary? Если не приписать каждому параграфу уникальный класс, то единственный способ решить эту задачу состоит в применении синтаксиса селекторов, появившегося в CSS2:

```
#quickSummary > p + p {
 color: red;
}
#quickSummary > p + p + p {
 color: inherit;
}
```

Поясним: первое правило содержит комбинацию селекторов дочерних и сестринских элементов и означает: «Применить это правило к любому элементу p, которому непосредственно предшествует другой элемент p и который также является дочерним элементом #quickSummary». Иными словами, это все абзацы внутри #quickSummary, кроме первого. Поэтому второе правило вводит дополнительное условие, отменяя применение первого ко всем абзацам, перед которыми имеется не менее двух предшествующих абзацев, из которых первый должен быть дочерним элементом #quickSummary; иначе говоря, для всех абзацев, кроме первых двух, восстанавливается унаследованный стиль.

Уф! Пожалуй, слишком много путаницы ради простого выбора элемента. К счастью, в CSS3 это можно сделать гораздо проще:

```
#quickSummary p:nth-child(2) {
 color: red;
}
```

Но ни один из этих способов не работает в Internet Explorer! Так что можете на время забыть о них, если только нет жесткой необходимости использовать стиль, который большая часть аудитории не увидит. Конечно, в течение следующих нескольких лет предпочтения публики могут измениться, но пока Internet Explorer занимает примерно 90% рынка.

Разговор о сложных селекторах в CSS увел нас довольно далеко от простой задачи. Но теперь должно быть ясно, зачем нужны дополнительные классы: они обеспечивают максимальную гибкость дизайна. Причины для включения в разметку «лишних» элементов span такие же: наличие внутренних элементов в каждом абзаце расширяет возможности применения стилей.

СОВЕТ

Подробнее о селекторах в CSS2 можно прочитать на сайте W3C в статье «5 Selectors» (www.w3c.org/TR/CSS21/selector.html).

СОВЕТ

Подробнее о селекторах в CSS3 можно прочитать на сайте W3C в статье «Selectors» (www.w3c.org/TR/css3-selectors).

СОВЕТ

Прекрасным инструментом для расшифровки сложных селекторов CSS является SelectORacle компании OPAL Group (http://gallery.theopalgroup.com/ selectoracle).

Подстановка изображений

У элементов span есть и еще одно назначение. Когда проектировался сайт Zen Garden, умами дизайнеров во всем мире как раз овладела совершенно новая идея: если имеется заголовок (скажем, h3) и вы хотите заменить текст в нем изображением, то этот текст необходимо окружить двумя элементами. Один из них h3, а другой - span.

Если скрыть текст внутри span, например, за счет правила display: none, то останется только h3, с которым можно связать фоновое изображение, создав тем самым иллюзию графического заголовка. Поскольку у каждого элемента в Zen Garden имеется внутренний span, то открываются широкие возможности для экспериментов.

А зачем вся эта суета, спросите вы? Разве с помощью элемента img нельзя сделать то же самое? Ответ все тот же: гибкость. Радикально изменить внешний вид страницы за счет подмены таблицы стилей не удалось бы, будь изображения «зашиты» в HTML-код. Впрочем, у техники подстановки изображений есть свои проблемы, о которых мы еще поговорим ниже.

ПРИМЕЧАНИЕ

Как быть, если целью дизайна как раз является обеспечение гибкости? В некоторых проектах действительно необходима гибкая стилизация, и довольно часто задают вопрос, как максимально полно задействовать потенциал стилей в собственной разметке.

Вполне можно начать с методов, применяемых в Zen Garden, если только пользоваться ими с умом. Мы рекомендуем для начала ограничиться набором самых основных элементов HTML и с их помощью описать контент: абзацы, заголовки, списки и т.п. По мере необходимости включите в стратегически важные места дополнительные контейнеры div с уникальными идентификаторами id и добавьте к некоторым элементам имена классов. Встречающиеся в Zen Garden элементы span, наверное, вам не понадобятся, пользуйтесь ими только, когда возникнет необходимость.

Упрощение разметки

Некоторые решения принимались, исходя из знаний, имевшихся на тот момент. В результате в разметку было введено больше структурных элементов, чем необходимо в свете появления новых приемов. Нетрудно найти места, откуда можно было бы убрать лишние элементы.

С момента запуска Zen Garden стало возможным подставлять изображения без дополнительного span, так что большинство этих элементов больше не нужны. И избыточные классы не используются в 98 процентах присланных работ, так что и от них можно было бы с легкостью отказаться. Набор пустых div и span в конце документа нужен только ради дополнительных адресуемых элементов, которыми дизайнер мог бы воспользоваться.

Очень может быть, что для других сайтов такие элементы не понадобятся. В Zen Garden на них возложена весьма специфическая функция – обеспечить максимальную гибкость. Но настолько вариативный дизайн нужен далеко не каждому сайту.

Если вы захотите воспользоваться HTML-разметкой Zen Garden (а мы не возражаем против этого), то не забывайте, что одна из важнейших целей дизайна на

основе стандартов – устранить лишние элементы. Включить один-другой `div` для обрамления логически связанных элементов вполне допустимо, но окружать `span` каждый абзац и каждый заголовок на странице, пожалуй, чересчур. Ниже в этой главе мы еще вернемся к обсуждению вопроса о структурировании документа.

Что мы уже знаем

Главная миссия сайта Zen Garden – продемонстрировать, на что способен дизайн на основе стандартов; знакомство с различными дизайнами показывает, насколько гибкие возможности представления дает CSS. Сам по себе CSS – это язык дизайна, служащий дополнением к таким языкам разметки, как HTML, XHTML и XML. Спецификация CSS1 появилась в 1996 году, а CSS2 – в 1998. В браузеры постепенно вводится поддержка для той и другой, так что начиная с 1998 года стало принято управлять всеми типографскими особенностями представления с помощью внешнего CSS-файла. И на этом роль CSS и ограничивалась в течение последующих четырех или пяти лет.

Легко понять, почему CSS использовался лишь для управления шрифтами – ведь только это и поддерживалось одинаково всеми браузерами. Впрочем, даже и с этим они справлялись не очень хорошо, но у такого способа было одно весомое преимущество, из-за которого большинство дизайнеров готовы были применять CSS несмотря ни на что.

Преимущество состояло в том, что все документы на сайте могли ссылаться на один и тот же CSS-файл. Если задать гарнитуру шрифта в таблице стилей, то соответствующее объявление будет применено ко всем страницам. Стоит изменить вид или размер шрифта всего в одном файле, как это отразится на всем сайте. Это было существенно лучше применявшейся ранее методики, когда тег `<font>` приходилось ставить перед каждым нуждающимся в выделении элементом документа на всех страницах сайта. В результате на каждой из сотен страниц могло оказаться по семь-восемь таких тегов. Изменение шрифта выливалось в длительную и мучительную работу.

Этот тип отделения представления был лишь малой толикой того, на что способны таблицы стилей. Редизайн сайта стал на много порядков проще, поскольку для переноса текста из старого дизайна в новый уже не нужно было выискивать и заменять тысячи тегов `<font>`, разбросанных по разным файлам; достаточно было скопировать контент, а таблицы стилей сделали бы все остальное.

Но долгое время этим все и ограничивалось. Старые браузеры, плохо поддерживающие CSS и не позволявшие применять более сложные стили, оставались на компьютерах пользователей в течение многих лет после предполагаемого срока замены, так что делать что-то более продвинутое просто не имело смысла.

Наконец в начале нового тысячелетия поддержка со стороны основных браузеров улучшилась до такой степени, что можно было всерьез рассматривать задействование потенциала CSS в полной мере. Конечно, она была (да и до сих пор остается) несовершенной, но все же достаточна, чтобы применение CSS не сводилось к замене шрифтов. После нескольких лет интенсивного сотрудничества веб-дизайнеры во всем мире выработали надежные методы работы с имеющимися стилевыми средствами, позволяющими верстать страницы с использованием плавающих элементов и модели позиционирования. Были найдены и обходные пути для браузеров, не вполне соответствовавших стандартам.

Так где же мы сейчас находимся? Даже в 2005 году еще есть браузеры, не до конца поддерживающие спецификацию CSS2, хотя она вышла в свет в 1998 году. Но в общем и целом поддержка достаточна для того, чтобы достоинства CSS и дизайна на основе стандартов стали осязаемы и начали приносить плоды.

Приглядимся к этим достоинствам внимательнее. Скорее всего, вы о них уже знаете, коль скоро читаете эту книгу, но если вас интересуют детали, советуем обратиться к источникам, упомянутым в разделе «В заключение» в конце книги.

Достоинства стандартов Web

Говоря о стандартах Web, следует представлять себе целиком всю картину, включая как CSS, так и язык HTML, применяемый для структурной разметки. Трудно перечислить достоинства одной технологии без упоминания другой. Выигрыш получается лишь тогда, когда обе они применяются совместно: содержание и структура документа описываются с помощью HTML, а дизайном управляют таблицы стилей.

Варианты дизайна, размещенные на сайте Zen Garden, – доказательство того, что один и тот же HTML-файл можно визуализировать бесчисленным числом способов, если применить к нему разные таблицы стилей. Технология CSS – это метод представления документа путем применения различных стилей к его структурным элементам. Хотя он и применяется к структуре, но работает независимо от нее, позволяя создавать различные визуальные образы документа.

К исходному документу предъявляется лишь одно требование: он должен быть правильным и достоверным. Неправильно размеченные документы будут прорисовываться непредсказуемо, что вызовет массу проблем. Поэтому при поиске причин ошибок в результате применения CSS первым делом проверьте, что разметка успешно проходит контроль на правильность.

ПРИМЕЧАНИЕ

Программа контроля правильности HTML-разметки находится на странице Validate Your Markup сайта W3C (http://validator.w3.org).

Файлы меньшего размера загружаются быстрее

Когда средства представления выносятся из разметки в таблицы стилей, размер файла обычно уменьшается. 30 Кбит, уходивших ранее на разметку, вполне могут превратиться в 10 Кбит на HTML плюс 10 Кбит или меньше на CSS.

Помимо уменьшения размера самого файла, есть и еще один источник оптимизации – CSS кэшируется в памяти браузера. Судите сами: когда HTML-разметка занимала 30 Кбит, при каждой загрузке страницы необходимо было передать эти 30 Кбит плюс собственно контент документа. При просмотре десяти страниц сайта приходится без нужды загружать целых 300 Кбит.

Если же вынести дизайн во внешний CSS-файл, то, во-первых, уменьшится размер загружаемой страницы, а во-вторых, таблицы стилей не придется грузить при просмотре каждой страницы. Останется лишь 10 Кбит HTML-разметки на страницу, поскольку остальные 10 Кбит, приходящиеся на CSS, будут извлекаться из кэша браузера. Таким образом, вместо вышеупомянутых 300 Кбит накладных расходов нужно будет загрузить всего 110.

ПРИМЕЧАНИЕ

Соображения о повышении пропускной способности применимы только в случае, когда таблицы стилей выносятся в отдельный файл; можно разместить их и в самом HTML-документе, но тогда, в отличие от внешнего CSS-файла, они не будут кэшироваться, так что часть преимуществ будет утрачена.

Такое уменьшение «веса» страницы за счет ее «разрезания» может положительно сказаться и на конечных пользователях, и на работе сервера. Если трафик велик, то сокращение числа передаваемых байтов может весьма заметно сэкономить полосу пропускания. Веб-сайт спортивной сети ESPN перешел на CSS в начале 2003 года, и приводимые цифры звучат просто фантастически. В интервью, которое впервые было опубликовано на несуществующем ныне сайте DevEdge компании Netscape (его можно найти по адресу www.mikeindustries.com/blog/archive/2003/06/espn-interview) разработчики ESPN.com сообщают, что *ежедневная* экономия составила 2 терабайта. А это значит, что теперь сайт может обслуживать больше посетителей и делать это быстрее – только за счет изменения технологии разметки. Неплохая отдача на единовременные затраты.

Большая переносимость

Вообще-то для прорисовки правильно структурированной страницы даже и таблица стилей не нужна. Иногда это имеет прямой смысл – рассчитывая на большой монитор с разрешением 1024×768 пикселей, вы имеете в своем распоряжении достаточно места, чего не скажешь, когда речь идет о карманном компьютере с экраном 240×320 пикселей.

Без CSS «голый» контент вполне можно читать, ссылки работают, и все элементы формы функционируют как положено. Так что, ограничившись структурной разметкой и больше ничего не делая, вы автоматически получаете версию своего сайта для мобильных устройств.

Но CSS предлагает поддержку для разных типов устройств и позволяет точно указать стиль, применяемый для вывода на конкретное устройство. Компьютерному монитору, мобильному устройству (скажем, сотовому телефону) и принтеру соответствуют разные типы устройств. На самом деле их даже больше, причем некоторые еще только разрабатываются.

ПРИМЕЧАНИЕ

Типы устройств, поддерживаемые CSS, описаны на странице «7 Media Types» сайта W3C (www.w3.org/CSS21/media.html).

Таблицы стилей для печатных устройств выносят смертный приговор ссылочке Print This Page (Распечатать страницу). Любую страницу сайта можно распечатать, и для этого нужна лишь простая таблица стилей, которая форматирует одни элементы, прячет другие и строит макет с учетом особенностей принтера. Поскольку принтеры накладывают иные ограничения, нежели экран, то тонкий контроль, предлагаемый таблицами стилей, весьма желателен.

Теоретически когда-нибудь для каждого устройства вывода будет определен подходящий тип. На практике же сейчас найдется мало людей, пользующихся для просмотра веб-страниц чем-нибудь, кроме экрана (не исключая авторов страниц и производителей оборудования). Реализуемые с помощью CSS средства управления отображением на экранах сотовых телефонов редко заходят дальше настройки шрифтов и цветов, а браузер для устройства типа aural (голосовое воспроизведение) еще не написан.

Это классическая проблема курицы и яйца: поддержка не улучшится, пока не появится контент, но никто не станет готовить контент для несуществующего устройства. Однако хотя и медленно, но изменения происходят, так что, возможно, мы еще увидим вариант Zen Garden, стилизованный для сотового телефона или КПК.

Улучшенная доступность

Правильно структурированная HTML-разметка доступна уже на стадии проектирования; вы можете создать версию сайта, на 90 процентов удовлетворяющую рекомендациям W3C по доступности контента, просто за счет корректной разметки.

> **ПРИМЕЧАНИЕ**
> Рекомендации по доступности веб-контента находятся по адресу www.w3.org/TR/WAI-WEBCONTENT.

Попрощайтесь с текстовыми вариантами контента и несколькими версиями сайта, которые так трудно синхронизировать; им на смену приходит единственный вариант верстки. В большинстве случаев даже пользователи с ослабленным зрением, которым приходится пользоваться программами, воспроизводящими содержимое веб-страницы голосом, смогут воспользоваться вашим сайтом – безо всяких усилий с вашей стороны.

Тут, конечно, есть и подводные камни, и мы немного поговорим о них в следующей главе, когда будем обсуждать, какие проблемы с точки зрения доступности может создать применение CSS.

Точный контроль

Будучи языком стилизации, CSS предлагает широкое разнообразие способов форматирования, которые в HTML были попросту невозможны: точное задание интерлиньяжа (расстояния между строками текста) с помощью свойства `line-height`, расстояния между элементами с помощью свойств `margin` и `padding`, позиционирование элементов на странице с помощью `absolute` и `relative` и многое другое.

Если вам всегда не нравилась сеточная структура, диктуемая таблицами, то возрадуйтесь: CSS-позиционирование позволяет избавиться от нее и располагать элементы в любом месте страницы. Если вам хотелось управлять фоном точнее, чем возможно с помощью одного лишь цвета и повторяющейся картинки, то свойство `background` и примыкающие к нему станет для вас маленьким открытием.

Чему мы научились

Не бывает идеальных сайтов, и Zen Garden – не исключение. Теперь уже поздно возвращаться к началу и что-то менять, но в ходе дискуссий на протяжении нескольких прошедших лет возникло новое понимание, которым хотелось бы поделиться.

Проверка доступности

Проверка составленной разметки с помощью программы Bobby показала, что большинству критериев доступности она удовлетворяет. Перед запуском сайта пришлось внести несколько мелких изменений, чтобы устранить замеченные Bobby недостатки. В хвостовик Zen Garden была добавлена метка «AAA» как свидетельство того, что о доступности позаботились.

А так ли это? Оказалось, что Bobby – еще не окончательный авторитет в вопросах доступности. Ознакомившись с «Рекомендациями по доступности веб-контента», опубликованными W3C (см. раздел «Улучшенная доступность» выше), вы поймете, что кое-что Bobby проверить просто не в состоянии. Так, пункт 2.1 гласит, что вся информация, доносимая с помощью цвета, должна быть доступна и при просмотре на устройстве, не поддерживающем цвета. Но программа типа Bobby не может этого подтвердить, особенно если цвет является частью графического изображения.

ПРИМЕЧАНИЕ

Bobby (http://bobby.watchfire.com) – онлайновый инструмент проверки доступности. Есть и другие, например Cynthia Says (www.cynthiasays.com) и WebAIM's Wave (http://wave.webaim.org).

Вообще, если внимательно изучить сайт Bobby, то обнаружится следующий текст с отказом от ответственности: «Доступность – это в конечном итоге вопрос человеческих способностей. Она определяется тем, может ли группа людей с определенными ограничениями легко получить доступ к информации. Bobby – всего лишь попытка помочь разработчикам сделать веб-страницы более доступными, но полной доступности эта программа гарантировать не может».

Часто задаваемые вопросы: http://bobby.watchfire.com/bobby/html/en/faq.jsp#faq_onetool.

Таким образом, сайт Zen Garden прошел все проверки доступности, относящиеся к HTML-разметке, но существуют аспекты, с HTML не связанные. Некоторые относятся и к CSS, и со временем стало ясно, что в некоторых вариантах дизайна они не приняты во внимание. Когда в главе 4 мы будем обсуждать метод подстановки изображения Фарнера, то увидим, как легко с помощью CSS создать проблемы, которые никакой автоматический инструмент не обнаружит.

Урок таков: автоматические средства проверки типа Bobby могут служить в качестве отправной точки для создания доступных сайтов, но в «Рекомендациях» есть много не менее важных пунктов, которые с их помощью проверить нельзя.

Масштабируемость текста

Одна из проблем CSS, относящаяся к доступности, – что делать при изменении размера текста. Очень хочется выбрать шрифт поменьше, чтобы на экране поместилось больше информации. Все мы знаем, как мало места на экране, так что основания для такого решения имеются.

Проблема в том, что многие люди просто не могут читать текст, набранный слишком мелким шрифтом. Для некоторых нечитаемым является шрифт высотой 12 пикселей, а иногда даже 24 пикселей недостаточно. Разумеется, учитывая невысокую разрешающую способность современных мониторов, никто не станет делать дизайн со шрифтом высотой 24 пикселя, поэтому нужен какой-то компромисс.

К счастью, во всех современных браузерах имеется механизм изменения размера шрифта. Не все пользователи знают, как до него добраться, но для знающих его ценность неоспорима. Неприятность возникает в том случае, когда размер шрифта выражен в единицах px. Internet Explorer умеет масштабировать шрифты, если единица измерения em или %, но в случае px он бессилен.

Поэтому пользователь, увы, не может рассчитывать на помощь со стороны браузера, и проблема остается. Но ведь дизайнеру надо просто отказаться от единицы px, и все будет в порядке, не правда ли? Возможно. Однако в главе 5 обсуждается совсем другая проблема, которая может возникнуть в случае использования единиц em и %.

Но даже если текст успешно масштабируется во всех браузерах, это еще не все. Дизайн должен *учитывать* возможность масштабирования (рис. 15). На первый взгляд, ничего сложного, но если вы будете перемежать элементы фиксированного размера (скажем, фоновую картинку высотой 20 px) с элементам переменного размера (например, текст, высоту которого браузер изменил с 14 до 24 пикселей), то быстро обнаружите, что макет начинает «плыть», когда размер шрифта отличается от того, что ожидал дизайнер. Большинство вариантов дизайна Zen Garden протестировано с этой точки зрения и может масштабироваться на размер, несколько больший принятого по умолчанию.

Важно понимать, что какой бы размер шрифта вы ни выбрали для текста, всегда найдется некто, желающий его увеличить. Нужно помнить о том, с какими проблемами придется столкнуться, когда текст становится крупнее, и учитывать их при верстке страницы.



Рис. 15 ✦ Текст масштабируется, а изображения и пропорции – нет

ПРИМЕЧАНИЕ

На сайте A List Apart размещена дополнительная информация о XHTML и различных значениях DOCTYPE (www.alistapart.com/articles/doctype).

XHTML и типы MIME

Сайт Zen Garden в момент запуска соответствовал спецификации XHTML 1.1. В конце концов, раз уж разметка прошла проверку на совместимость с XHTML 1.1, то почему бы не использовать ее вместо более старой спецификации XHTML 1.0 Strict?

Кстати говоря, XHTML – это не просто требование правильно закрывать теги. Даже если при проверке оказывается, что XHTML-код абсолютно корректен, то технически страница не будет считаться XHTML-документом, если для нее неправильно указан тип MIME.

MIME (Multipurpose Internet Mail Extensions – многоцелевые расширения электронной почты) – это стандарт для описания информации, и единственная причина, по которой вы должны о нем знать, состоит в том, что правильный тип MIME для XHTML-документов – это `application/xhtml+xml`. Можно указывать и гораздо более распространенный тип `text/html`, но только в контексте спецификации XHTML 1.0 Transitional или в том случае, если вы хотите, чтобы документ, удовлетворяющий спецификации XHTML 1.0 Strict, был обратно совместимым. Возвращать документ, составленный в соответствии со спецификацией XHTML 1.1, как тип `text/html` нельзя, ему обязательно должен быть приписан тип MIME, говорящий, что это XML-документ.

> **ПРИМЕЧАНИЕ**
>
> Парочку статей, в которых типы MIME описываются более подробно, можно найти на сайтах Web Standards Project (www.webstandards.org/learn/ askw3c/sep2003.html) и XML.com (www.xml.com/pub/a/2003/03/19/dive-into-xml.html).

Задание нужного типа MIME для возвращаемого документа сводится к необходимости изменения формируемых сервером заголовков, например, с помощью конфигурационных файлов `.htaccess` или `httpd.conf`. Если вы не понимаете, о чем речь, это не ваша вина. Для непосвященных это слишком технический вопрос, имеющий отношение к администрированию сервера.

Но даже выяснив, как задать правильный тип MIME, вы еще отнюдь не решили всех проблем. Только самые последние версии браузеров поддерживают тип `application/xhtml+xml`, поэтому нужен какой-то способ вернуть им и только им документ правильного типа, а для всех остальных указать более старый и привычный тип MIME.

Далее, коль скоро вы укажете правильный тип MIME, браузер вызовет для вашего документа синтаксический анализатор XML. А в XML есть правило, гласящее, что при обнаружении любого несоответствия, например отсутствующего тега </p>, следует прекратить разбор и вернуть сообщение об ошибке. Тогда вы не увидите в браузере даже частично прорисованного документа (рис. 16). XML-документ обязан быть правильным, и с этим ничего не поделаешь – ошибки не прощаются.

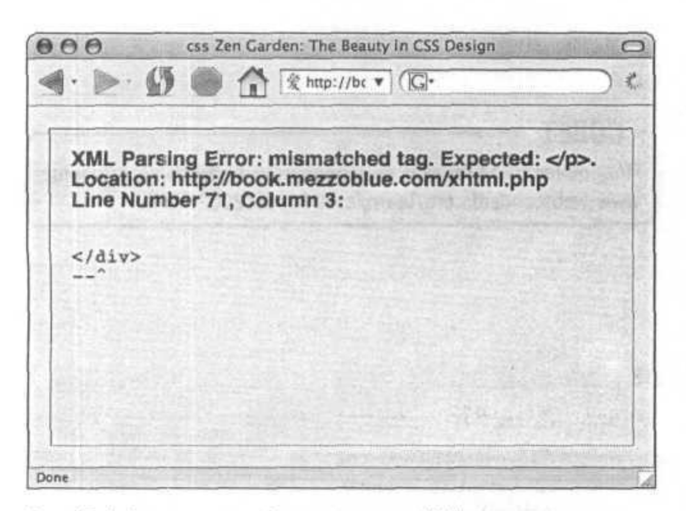


Рис. 16 ✦ Результат ошибки в структуре XML-документа

Стоит ли XHTML таких сложностей? Но ведь суть XHTML как раз в том и состоит, чтобы преобразовать HTML в одно из приложений XML. Поэтому некоторые считают странным использовать XHTML и не сообщать браузеру, что это настоящий XML-документ. Преимуществ у такой «честности» почти нет, но, если быть педантом, то это необходимо. Если вы не хотите связываться со всеми этими проблемами, то вы не одиноки. Многие широко известные в мире дизайнеры, ни секунды не колеблясь, отправляют XHTML-документы, описывая их типом `text/html`. Так что вы оказались в неплохой компании; вот только избегайте XHTML 1.1, придерживаясь одного из вариантов спецификации XHTML 1.0. Можно и вообще уйти от этой проблемы, оставаясь на уровне HTML 4.01, сопоставляя документу тип `text/html`, но при этом соблюдая правила XHTML.

Теперь вы знаете, что XHTML прост только на первый взгляд, а при попытке использовать его в реальных приложениях вас поджидает масса неприятностей.

Иностранные языки

Было время, когда написать на компьютере документ не на английском языке (в предположении, что живете вы в англоговорящей стране) было почти невозможно без специальных программ.

К счастью, нынешнее программное обеспечение с пониманием относится к существованию других языков. Лучшим подтверждением тому может служить Всемирная паутина, в которой число неанглоязычных сайтов стремительно растет.

Чтобы расширить аудиторию, Zen Garden обратился к добровольцам с просьбой перевести сайт на другие языки (*www.mezzoblue.com/zengarden/translations*). По мере появления переводов обнаружились новые проблемы. Для работы с другими языками нужно иметь некоторое представление о кодировках символов. Кроме того, перевод – это скорее искусство, чем наука.

Кодировки символов

Современные операционные системы прекрасно умеют работать с неанглийскими символами. Базовая инсталляция Windows XP или Mac OS X не всегда включает весь набор естественных языков, но на установочном диске имеются языковые пакеты, позволяющие отображать многие иностранные символы.

СОВЕТ

Подробнее о кодировках символов можно узнать на странице www.webstandards.org/learn/askw3c/dec2002.html.

Однако возможность видеть иностранные шрифты – это всего лишь половина дела. Копирование и вставка текста из какого-либо источника в HTML-документ будет работать, только если кодировки совпадают. Так, текст в кодировке shift_jis (японский) не будет правильно отображаться в документе с кодировкой utf-8 (Unicode). Важно убедиться в том, что кодировка HTML-документа соответствует тому тексту, с которым вы работаете.

Кодировку можно задать на уровне сервера для всех страниц сайта. Но даже в этом случае следует указывать кодировку в самом документе с помощью тега meta:

```
<meta http-equiv="content-type" content="text/
html; charset=iso-8859-1"/>
```

На сайте Zen Garden применяется кодировка iso-8859-1, стандартная для большинства европейских языков. В переводах используются другие кодировки, и, как мы теперь понимаем, правильнее всего было бы во всех случаях пользоваться кодировкой UTF-8.

UTF-8 – это вариант популярной международной кодировки Unicode. Преимущество ее в том, что в одном документе могут сосуществовать различные наборы символов, например из французского, японского, арабского и греческого языков.

Единственное препятствие на пути широкого распространения этой кодировки в Web состоит в отсутствии поддержки со стороны устаревших серверных программ и редакторов, не рассчитанных на многоязычность. У современных браузеров таких ограничений нет, так что пользователи никаких проблем испытывать не будут, все дело в сервере. Кроме того, при работе с азиатскими языками использование UTF-8 приводит к созданию файлов большего, чем реально необходимо, размера из-за сложного набора символов, поэтому если документ написан в основном на китайском, то лучше применять менее универсальную кодировку.

Итак, благодаря поддержке нескольких языков кодировка UTF-8 является предпочтительной для использования в Web при наличии соответствующих инструментов.

Расхождения в переводе

Поскольку переводом Zen Garden занимаются добровольцы, качество не всегда одинаково. Носители языка часто присылают замечания с предложениями, как исправить ошибки; очевидные грамматические и орфографические ляпы устраняются легко, но вот с заменой слов возникают проблемы.

Тут даже профессиональные переводчики не помогут, ведь человеческие языки не взаимозаменяемы. Одну и ту же фразу можно перевести по-разному в зависимости от опыта переводчика и культурного контекста. Например, когда мы говорим «That will never fly», имеется ли в виду неудачная идея или неисправный самолет? Enlightenment (нирвана или облегчение) – это идея дзен-буддизма или методика избавления от лишнего веса? Контекст может устранить двусмысленность, но только в том случае, когда он правильно понят.

Человеческий язык не такой точный, как компьютерный, поэтому от этой проблемы никуда не деться. Лучшее, на что можно рассчитывать, – это согласие между многими носителями языка. Но примите во внимание местные диалекты и различия, накопившиеся с течением времени (американцы, например, пишут color, а англичане – colour), и станет понятно, что даже такое согласие достижимо с трудом. Особенно если вы не говорите на соответствующем языке!

Таким образом, перевод документа – непростая задача, и достичь совершенства тут никогда не удастся. Поэтому на сайте Zen Garden мы с легким сердцем поместили заявление о том, что ошибки возможны, и мы готовы с ними смириться.

Авторское право и воровство

Сайт Zen Garden – не только выставка, но еще и источник вдохновения и знаний. Однако это палка о двух концах: стоит поделиться своими знаниями, как обязательно найдется кто-то, желающий получить больше, чем вы согласны отдать.

С самого начала было объявлено, что все присылаемые работы подпадают под условия лицензии Creative Commons License, разрешающей ограниченное использование содержащихся в них таблиц стилей. Но поскольку уникальность работы в не меньшей степени обусловлена графикой, то полное авторское право остается за дизайнером. Любому человеку, желающему включить элементы представленного дизайна в свои проекты, разрешено копировать и модифицировать CSS, но заимствовать GIF, JPG или PNG-файлы по отдельности или целиком можно только с явного согласия автора.

Теоретически этого казалось достаточно, чтобы сохранить авторские права на оригинальную работу, не препятствуя использованию ее в качестве образца и заимствованию частей CSS-файлов. Тем самым лишь поощрялось стремление изучить новую технологию.

Цель сайта – делиться знаниями, так что вроде бы описанные условия обеспечивали приемлемый компромисс. На практике все оказалось далеко не так идеально.

Вскоре после запуска Zen Garden стали появляться первые подражания. Некоторые «тянули» дизайн целиком, включая всю графику. Это очевидное нарушение авторских прав, поскольку все изображения защищены.

Другие слегка модифицировали CSS-файлы и графику либо вставляли на свои сайты элементы дизайна так, что назвать это лишь повторным использованием можно было с большой натяжкой. В некоторых случаях окончательный результат не вызывал слишком тесных ассоциаций с Zen Garden. Но иногда он так походил на источник, что автор оригинала бывал недоволен.

Что можно с этим поделать? Как правило, бывает достаточно послать электронное письмо, чтобы уладить проблему, поскольку большинство людей все же уважают авторские права и пожелания человека, от которого они черпают вдохновение. Но если нарушитель не желает идти навстречу, то принудить его нелегко. Если оригинальная графика не выставлена на сайте, то условия лицензии Creative Commons не позволяют привлечь к ответственности. Технически такое повторное использование допустимо, хотя многим авторам это не нравится.

Нарушение авторских прав на произведения искусства всегда было предметом споров. Насколько сильно надо изменить оригинал, чтобы результат можно было счесть новым произведением? Разрешено ли взять за основу чужую защищенную авторским правом работу, если конечный результат будет полностью отличен от нее? На такие вопросы даже суды затрудняются дать определенный ответ.

Для сайта Zen Garden идеальным является решение, позволяющее использовать таблицы стилей в качестве образца, но запрещающее полное или частичное копирование дизайна для коммерческих (или иных) целей без согласия автора. Грань здесь довольно тонкая: если лицензия излишне открыта, то авторы не захотят представлять свои работы; если чересчур закрыта, то ценность Zen Garden будет в значительной мере утрачена.

Если не закрывать сайт, то никогда не будет гарантий от заимствования помимо желания автора. Есть лицензия или нет, но таково свойство любой опубликованной творческой работы: ее воруют. Это происходит неизменно, загляните на сайт pirated-sites.com.

ПРИМЕЧАНИЕ

На CSS-файлы распространяется лицензия «с упоминанием автора, для некоммерческого использования, с теми же условиями распространения своих работ». Подробнее о ее условиях можно прочитать на сайте Creative Commons (http://creativecommons.org/licenses/by-nc-sa/2.0).

Причем в Web украсть проще благодаря возможности просмотра исходных текстов, имеющейся в любом браузере. Правда, и поймать вора проще – ведь есть протоколы обращений и поисковые машины.

Во время работы над этой книгой проблема была еще далека от решения. Было сочтено разумным все же разрешить ограниченное использование CSS-файлов на условиях лицензии Creative Commons с запретом применения в коммерческих целях. Такое решение не идеально, но работает в предположении, что у большинства людей все же добрые намерения.

В общем, мы поняли, что урод в семье обязательно найдется, какой бы лицензией ни защищать свою работу, так что нужно лишь выбрать метод извещения добропорядочных граждан о том, что им разрешено делать.

Бери больше, кидай дальше!

Многие посетители нашего сайта предлагали, как сделать его больше, лучше, по-другому, громче, с витамином С и так далее. Предложений хватало на любой вкус. Вот перечень некоторых идей.

CMS Zen Garden

Применить стили к одной странице – это хорошо, но как насчет целого сайта? Почему бы не распространить концепцию на разные страницы настоящего, обновляемого сайта, сопровождаемого с помощью системы управления контентом (CMS), чтобы полнее отразить реальные условия?

E-commerce Zen Garden

Похожая идея – почему бы не сделать с помощью таблиц стилей электронный магазин с настоящей, постоянно обновляемой витриной? Стилизовать можно было бы корзину, товары, продвигаемые позиции и так далее.

JavaScript Zen Garden

Отделить структуру от представления – цель благородная, но почему бы не сделать еще один шаг и не ввести поведенческий уровень, отделенный от двух предыдущих?

Великолепную иллюстрацию этой идеи предложил один из ее авторов Рарес Портан (Rares Portan) (www.csszengarden.com/javascript/) (рис. 17), а обсуждение аргументов за и против можно найти на странице www.mezzoblue.com/archives/2004/05/ 04/javascript_b).

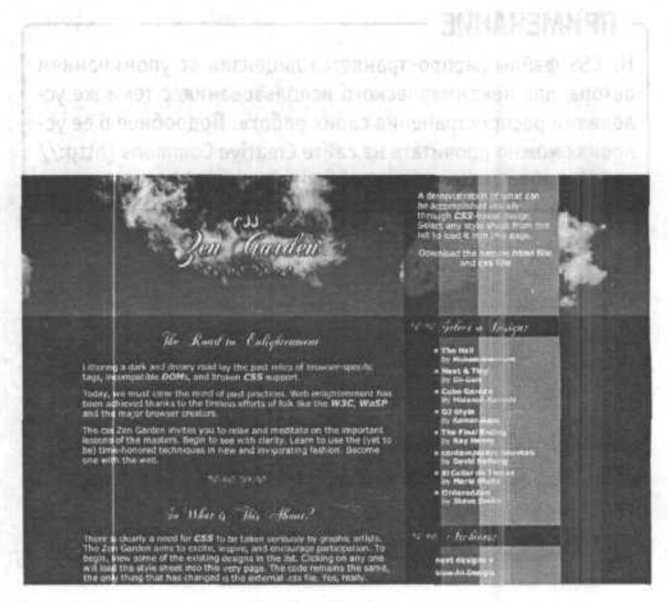
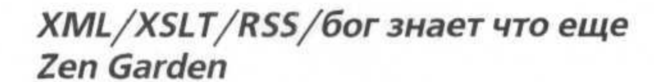


Рис. 17 ✦ Вариант дизайна Zen Garden со сценарием, реализующим плывущие облака



XML/XSLT/RSS/бог знает что еще Zen Garden

Подставьте любой желаемый язык разметки и примените к нему CSS-стили. Почему бы не продемонстрировать потенциал CSS в сочетании с другими языками на базе XML, а не только с XHTML?

И где это все?

Без сомнения, все эти идеи завораживают, и реализация многих из них принесла бы несомненную пользу веб-сообществу. В большинстве случаев авторы предложений получили положительные отклики и предложение разместить демонстрацию своих концепций вне сайта.

У сайта Zen Garden очень специфические цели, он не предназначен быть хранилищем всего для всех. Изменение правил игры и предназначения сайта сейчас, когда накоплено уже столько материала, было бы несправедливо по отношению к тем, кто прислал свои работы раньше.

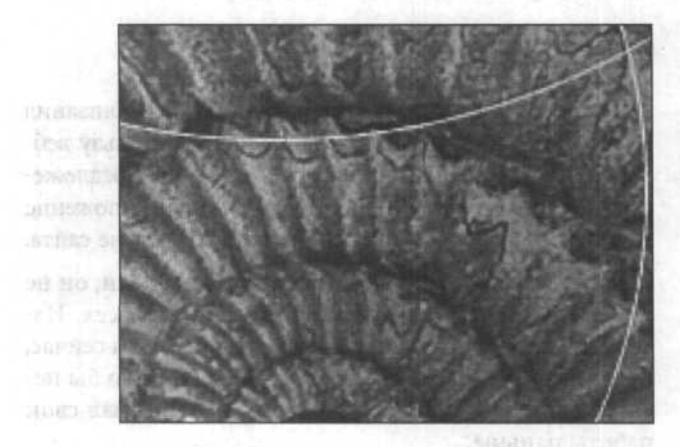
Но на базе предложенных идей можно организовать другие сайты. Если вы можете сделать лучше, приступайте. Один или два сайта даже были созданы, но долго не просуществовали. Есть разные мнения по поводу причин этого, мы поделимся своим.

Во-первых, на сайт Zen Garden очень *легко* присылать работы. Необходимо лишь быть искусным дизайнером и знать правила написания CSS, но на создание своего варианта дизайна не надо затрачивать несколько недель. Стилизовать целый сайт гораздо сложнее, особенно если контент постоянно изменяется. Если работа оплачивается, то оправдать временные затраты можно, иначе они становятся непреодолимым барьером.

Во-вторых, главная цель Zen Garden была достигнута. Мы пытались с его помощью изменить отношение к дизайну на основе CSS, мы это сделали. И хотя стили применяются всего к одной странице, но обобщить идею на целый сайт в принципе несложно.

Если кому-то абсолютно непонятно, как распространить концепцию на несколько страниц, то существует немало следующих стандартам сайтов, которые можно взять за образец. Многие корпоративные сайты перешли на дизайн на основе стандартов, например Wired News (www.wired.com), ESPN.com, Chevrolet (www.chevrolet.com), AOL.com и другие.

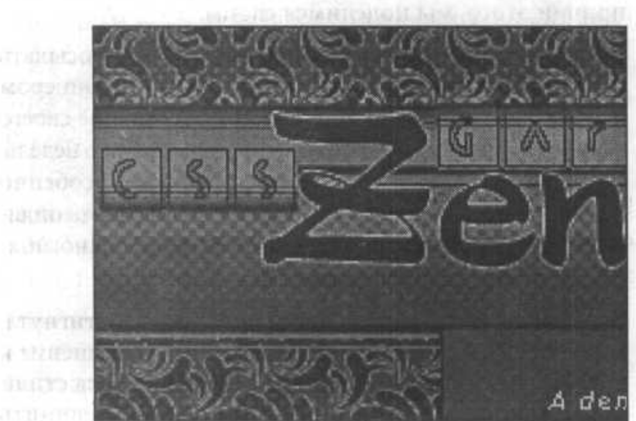
Таким образом, в самых разных отраслях имеется интерес к стандартам, а ценность последних реальные примеры доказывают даже убедительнее, чем чисто демонстрационный сайт типа Zen Garden.



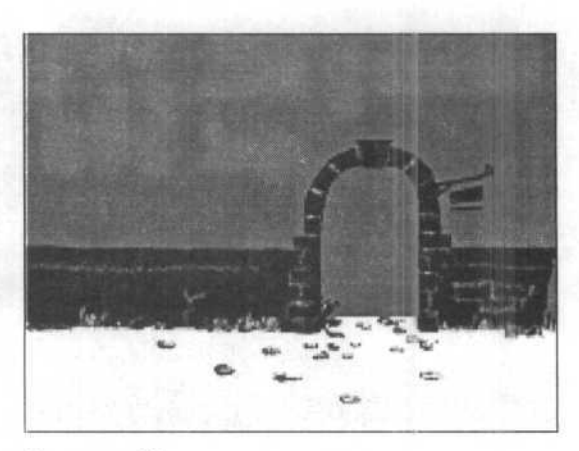
Атлантида, с. 54



Подсолнух, с. 60



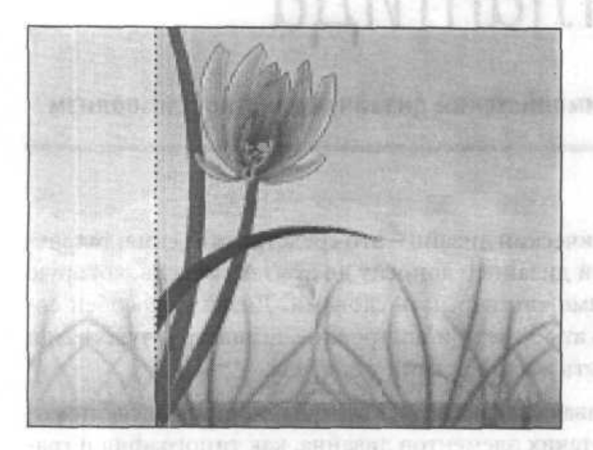
В зеленых тонах, с. 72



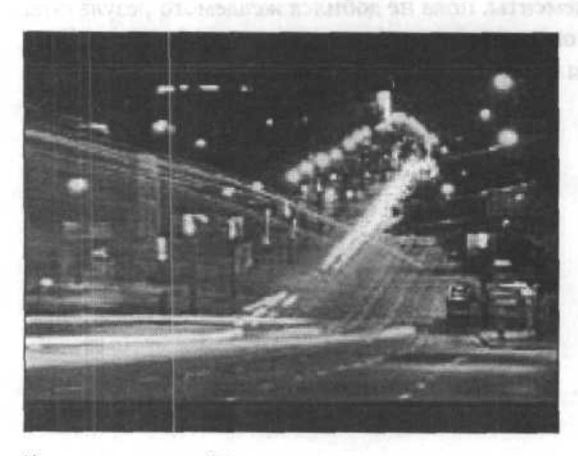
Баллада, с. 78

2

Дизайн



Весна, с. 66



Ночная поездка, с. 84

Процесс веб-дизайна претерпевает драматические изменения, но его принципы остаются прежними. Думаете ли вы о выборе палитры, визуальном потоке, об эффектном использовании образов или управлении визуальной близостью страничных элементов – традиционные ценности остаются в силе.

Обсуждаемые в этой главе варианты дизайна иллюстрируют эту точку зрения, отвечая одновременно на вопрос, все чаще встающий перед современными дизайнерами: как эффективно управлять технологическим процессом? Вне зависимости от того, что вы предпочитаете: работать в Adobe Photoshop, готовить эскиз, создавать прототип с помощью CSS или сразу приступать к кодированию, вы обязательно найдете в этой главе новые для себя полезные приемы.

Дизайнер **Кевин Дэвис** (Kevin Davis)

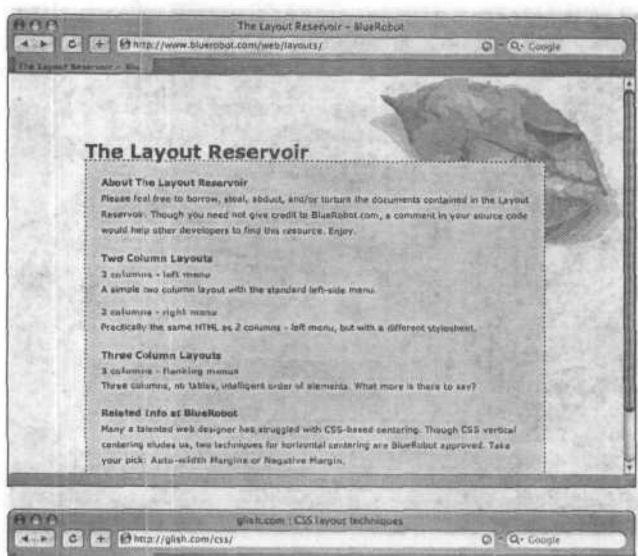
www.csszengarden.com/028

Атлантида

Минималистский дизайн, единство, символизм

Графический дизайн – это средство общения; талантливый дизайнер доносит до зрителя мысль, которую невозможно передать словами. Дизайн способен создать атмосферу и настроение, воззвать к чувствам и вызвать живой отклик у зрителя.

Создавая «Атлантиду», Кевин Дэвис стремился с помощью таких элементов дизайна, как типография и графика, проделать примерно то же, что делает дизайнер по интерьерам, подбирая сочетание столиков, стоек и красок для создания стильного кафе. Начав с выбора фотографий и цветовой палитры, Дэвис соединял разные элементы, пока не добился желаемого результата. Затем он наложил графический материал на заранее продуманный макет страницы.





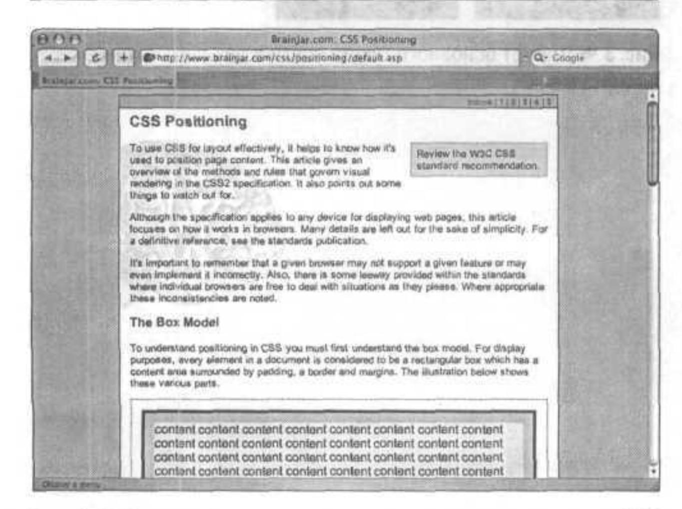


Рис. 1 ✦ Примеры ранних минималистских макетов на основе CSS

Минимализм

Первое, что бросается в глаза при взгляде на «Атлантиду», – это относительная простота. Исключение составляет разве что детальный коллаж в заголовке. Комбинируя части фотографий с обманчиво простой версткой, Дэвис достиг редкого сочетания сложности и минимализма.

Сам сайт CSS Zen Garden создавался в атмосфере, когда дизайн на основе CSS пренебрежительно отвергали, считая его «прямоугольным» и «слишком простым» (рис. 1). Ничего плохого ни в том, ни в другом подходе нет, если только они применяются в подходящем контексте.

Презрительное отношение к этим «типичным» CSS-дизайнам проистекало из непонимания простого факта: авторы не были графическими дизайнерами в привычном понимании этого слова, а вина возлагалась почему-то на технологию. Именно для исправления этой ситуации в первую очередь и задумывался сайт Zen Garden.

Но «Атлантида» дает убедительный аргумент в пользу того, что минималистский дизайн совсем не обязательно плох; ведь утомляют лишь бесчисленные повторения одного и того же. Все-таки картофельное пюре, жареная картошка и картофель фри – для одного обеда чересчур. Немного пюре (или соевого творога тофу, на ваш вкус) в качестве гарнира к индейке – более подходящее сочетание для простого, но вкусного обеда.

В ситуации, когда не нужны сложные зрительные образы, облегченный минималистский дизайн – как раз то, что надо. В конце концов, дизайн диктуется содержанием и, если содержанием визуально сложно для восприятия или требует повышенного внимания, то нет смысла перегружать его избыточным оформлением.

С другой стороны, создать по-настоящему *хороший* минималистский дизайн... Даже профессионалам это удается с трудом. Когда элементы дизайна просты, они привлекают больше внимания зрителя, чем надо бы. Каждый элемент становится важен, поэтому именно внимательное отношение к деталям отличает выдающуюся работу от посредственной.

Изучая блестящий минималистский дизайн «Атлантиды», вы поймете, как следует сочетать разные элементы. Если удалить заголовок с картинкой, то макет станет более понятным (рис. 2). Главная информационная область в середине соседствует с небольшим меню слева.

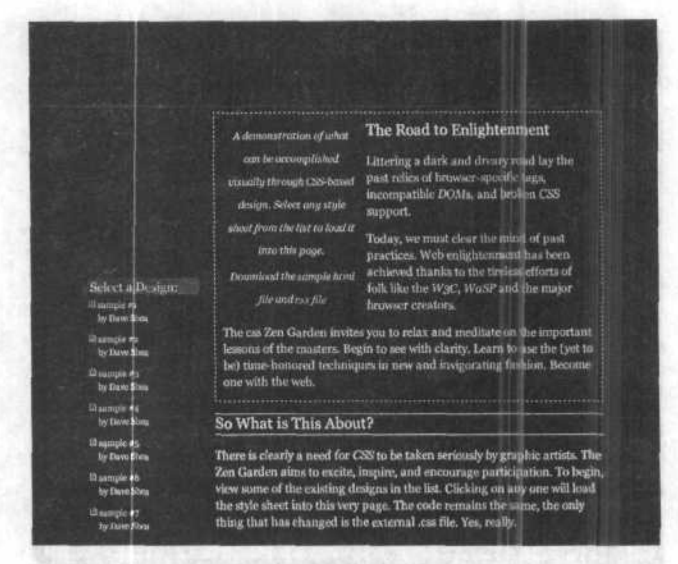


Рис. 2 ✦ Атлантида без заголовка с картинкой

Типография

Размер шрифта основного текста в дизайне «Атлантида» чуть больше размера шрифта в меню. Дабы подчеркнуть, что текст слева не так важен, как текст справа, Дэвис воспользовался типографскими свойствами (рис. 3). Кроме того, более крупный шрифт заголовка четко отделяет его от основного текста. Эти визуальные различия сообщают дополнительную информацию о сравнительной важности каждого фрагмента текста.

Вид шрифта – это способ передачи письменной речи, которой присущи определенная иерархия, построение и ритм. Акцентирование всего этого с помощью размера шрифта, интервалов и цвета – действенный способ сообщить читателю больше, чем написано.



Рис. 3 ✦ Шрифт основного текста почти в два раза крупнее шрифта меню

Иконография

В расположенное слева меню Дэвис поместил небольшие иконки напротив названия каждого варианта дизайна. Поскольку они расположены рядом со ссылками, то сливаются с текстом и выглядят как скромные маркеры, на которых глаз не задерживается. Эта мелкая деталь визуально отделяет название варианта от находящегося ниже текста. В результате простое, в сущности, меню выглядит более интересно.

Иконки (или пиктограммы) – это упрощенные символы, которыми можно представить как отдельный объект или идею, так и целую концепцию – путем замены длинного текстового описания визуальным образом. Визуально представить объект обычно просто, а вот визуализировать процессы или действия куда сложнее. Хорошая иконка не нуждается в пояснениях, но часто их приходится делать абстрактными. Надо стараться, чтобы вид иконки как можно точнее выражал ее функцию, чтобы после многократного просмотра у зрителя возникала устойчивая связь между тем и другим.

Линии

Дэвис выбрал два стиля для линий, разделяющих разные области. Пунктирной линией окружены верхние абзацы, накладывающиеся на картинку в заголовке, а сплошными белыми линиями отчеркнуты заголовки и хвостовики (рис. 4). Таким простым и ненавязчивым способом создается дополнительный визуальный контекст.

За счет одного лишь использования линий можно определить различные области на странице, установить границы, подчеркнуть порядок чтения материала и ввести дополнительное измерение. Но опасайтесь переизбытка линий: незамкнутые прямоугольники и изобилие параллельных отрезков оставляют впечатление, будто имеется некая табличная структура, которой на самом деле нет. Глаз пытается найти регулярность там, где ее, возможно, не существует, поэтому применяйте прямые линии только там, где это имеет смысл.

Рис. 4 ✦ Сплошные белые линии отделяют заголовки от основного текста

Поля

Сосредоточив основное содержание страницы в центральной колонке, Дэвис получил максимальную свободу в определении промежутков между элементами. За счет полей переменной ширины по обе стороны контент позиционируется в середине окна браузера. Иногда лучше выглядит дизайн, при котором заполняется все окно, а иногда занятую текстом область предпочтительнее ограничить.

Как бы то ни было, широкие поля слева и справа от основных текстовых областей создают впечатление простора. По умолчанию большинство браузеров оставляют очень узкие поля вдоль внешней стороны страницы, поэтому текст простирается почти от края до края окна.

Это тяжело для визуального восприятия. Отсутствие полей приводит к тому, что текст зрительно сливается с тем, что находится за пределами окна браузера. Технологические умолчания в этом случае выбраны неудачно, исправить ситуацию должен дизайнер.

В случае минималистского дизайна, когда поля являются одним из немногих использованных визуальных средств, от их правильного задания зависит очень многое. Широкие поля фокусируют взгляд читателя на том, что отображается в окне браузера.

ПРИМЕЧАНИЕ

Многие дизайнеры не считают плохой выбор стандартной ширины полей проблемой, поскольку все равно переопределяют ее, но для любителей правильные начальные значения важны. Никто не пострадает от того, что разработчики браузеров нормально «подготовят сцену», ведь, например, стандартный офисный принтер формирует достаточно широкие поля, чтобы не допустить перегрузки страницы текстом.

Единство и символизм

«Атлантида» – это прекрасный пример использования элементов дизайна для создания настроения, что и входило в намерения Дэвиса. Согласно древнегреческому философу Платону, Атлантида – это остров, погрузившийся в море много веков назад. Покрытая пятнами известкового налета спиральная раковина в сочетании с классическими статуями и колоннами в заголовке вызывает ощущение древности, сопутствующей античной легенде, а темно-синий и черный цвет наводят на мысль о морских глубинах. Заранее выбрав графические образы и цвета, Дэвис позаботился о том, чтобы дизайн отвечал его замыслу.

Согласованность

С помощью Adobe Photoshop цвета раковины были слегка изменены, а края немного сглажены, чтобы она сливалась с находящимся на заднем плане коллажем. Но Дэвис не стал смещать всю цветовую гамму раковину в сторону синих тонов, а для контраста оставил достаточно грязно-ржавого цвета (рис. 5). Применение в дизайне одной лишь синей гаммы было бы, наверное, насилием над зрителем (рис. 6).

Элементы дизайна могут варьироваться, но нужно стремиться сохранить единство цветового решения. Помимо излишне монотонной цветовой схемы, существуют и другие, более тонкие средства создать впечатление согласованности дизайна, например соответствие между добавленными линиями и фотографией или пропорциональность при выборе размеров шрифта и окружающих текст линий.

В заголовке «Атлантиды» мы видим сплошные белые дуги, пересекающие раковину и коллаж. Вне контекста они не вызывают никакого отклика. Но на синем фоне форма и цвет дуг очень напоминают рябь на воде. Их форма и размер соответствуют раковине, так что между двумя элементами образуется тесная связь.

Представление

Выбранный метод представления – простые элементы дизайна в сочетании со сложными образами и идеями – позволяют привнести в работу утонченность и глубину. Возможно, вы даже не замечали белых линий, пока мы не указали на них, но теперь, когда их



Рис. 5 ✦ Оранжевые тона раковины создают цветовой контраст



Рис. 6 ✦ Заголовок в одних лишь синих тонах больше соответствует теме, но ему недостает контрастности

ПРИМЕЧАНИЕ

Цветной вариант рис. 5 и 6 см. на вкладке.

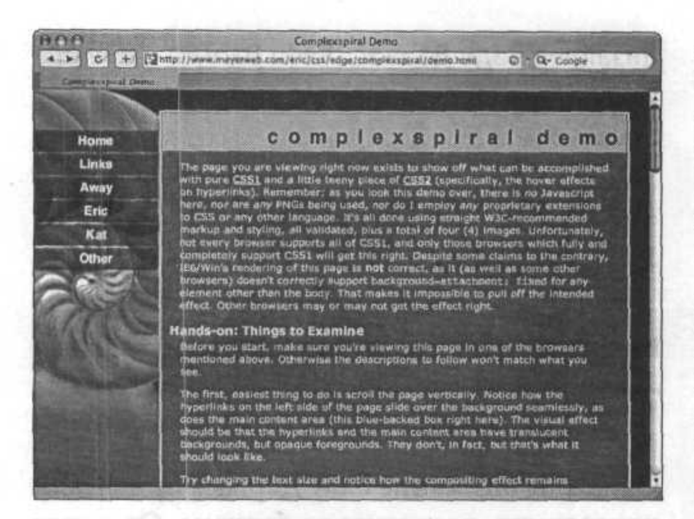


Рис. 7 ✦ Демонстрационная страница Complexspiral, на которую намекает раковина в заголовке «Атлантиды»

присутствие очевидно, трудно воспринимать их иначе, как рябь на поверхности воды.

Символизм «Атлантиды» не исчерпывается простыми линиями; следует еще обратить внимание на размещение графики внутри заголовка. Атлантида минувшей эпохи – могучая и богатая империя, символизируемая каменными колоннами и богато украшенными статуями, – изображена блеклыми тонами и помещена позади контрастной прорисованной Атлантиды наших дней: покрытой ржавчиной, древней и ушедшей в море. Пространственное расположение двух частей коллажа наводит на размышления о рухнувшей империи, дни славы которой давно миновали. Такое обманчиво простое решение стоит целого рассказа, зрителю есть над чем задуматься.

А для зрителя, разбирающегося в CSS, изображение в заголовке именно спиральной раковины имеет и еще один смысл: Дэвис тем самым отдал дань уважения классической странице, демонстрирующей CSS, а именно работе Эрика Мейера (Eric Meyer) Complexspiral, с которой можно ознакомиться по адресу www.meyerweb.com/eric/css/edge/complexspiral/demo.html (рис. 7).

Прерогатива дизайна

Дизайн – это мощное средство коммуникации; ясная и проводимая на всех страницах сайта мысль настраивает посетителя на определенный лад. Далеко не для всех сайтов необходимо повышенное внимание к дизайну; если центральное место занимают содержание и функциональность, то работа дизайнера должна оставаться незаметной и ограничиваться простой графикой и минималистской, но эффективной версткой.

В некоторых случаях дизайн должен содержать более явный посыл, зачастую даже не один. Не исключено, что при первом знакомстве некоторые тонкие задумки автора останутся непонятыми. Для интерпретации необходимы знания и опыт, поэтому, чтобы полностью проникнуть в смысл такой работы, как «Атлантида», необходимо попытаться встать на место дизайнера и понять его точку зрения.

Задача дизайнера – донести свое послание до максимально широкой аудитории, не утеряв при этом его неповторимости и убедительности. Но есть много способов скрыть за основным сообщением дополнительную информацию, что и доказывает «Атлантида» Кевина Дэвиса.



Дизайнер **Раду Дарвас** (Radu Darvas)
www.csszengarden.com/026

Подсолнух

Игра со светом и тенью, формой и пространством

Однажды внимание Раду Дарваса привлекли подсолнухи, растущие на балконе. Он начал снимать их, и в результате родилась идея дизайна «Подсолнух». Обработка фотографий в Adobe Photoshop помогла ему сформулировать идею работы для сайта CSS Zen Garden.

Прежде всего, он не хотел «прямоугольности». Но по мере продвижения увидел, что может управлять размещением колонок и создать у зрителя впечатление, будто они перекрываются. Удачные сочетания света и тени, формы и пространства привели к появлению яркого и энергичного «Подсолнуха».

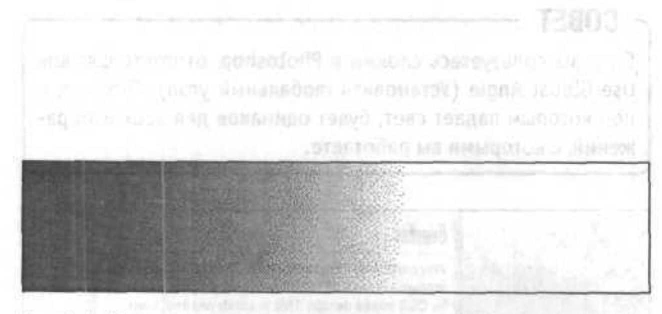


Рис. 1 ✦ Градиентная заливка прямоугольной области, датируется примерно 1995 годом. Сглаживание и разбиение на полосы при высоком коэффициенте сжатия выглядит очень непрофессионально

HOME

Рис. 2 ✦ Пример рельефного текста на рельефной кнопке. Такие визуальные эффекты широко использовались в веб-страницах в середине 90-х годов

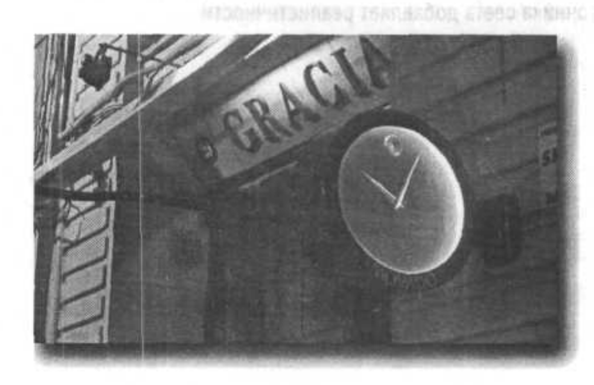


Рис. 3 ✦ Классический эффект падающей тени, примененный к фотографии

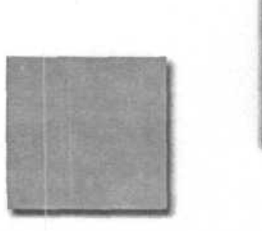




Рис. 4 ✦ Неправильное использование источника света – обратите внимание на различное направление теней

В поисках света

В то время как вы сидите и читаете эту книгу, рядом есть какой-то источник света. На секунду оторвитесь от чтения и поднимите глаза. Откуда исходит свет? Как он освещает близлежащие предметы? Куда падает тень?

Художники используют свет и тень, чтобы сделать картину более реалистичной, создать настроение и усилить впечатление от своей работы за счет интересных эффектов, вынесенных на передний план.

Оглядываясь на прошедшее десятилетие, мы найдем много применений свету в веб-дизайне. Тут и градиентная графика (рис. 1), и рельефные кнопки (рис. 2), и падающие тени (рис. 3).

К сожалению, веб-дизайнеры, не получившие надлежащего художественного образования, иногда пользуются светом неправильно. Необходимо учитывать ограничения «холста», с которым работаем. И не стоит жертвовать реалистичностью в угоду технике, применяя конфликтующие источники света (рис. 4).

Сохранение тени

Дизайн «Подсолнух» демонстрирует правильное применение источника света для создания тени. Хотя резкие переходы и неуместная рельефность в современном веб-дизайне уже вышли из моды, но падающие тени остались, и тому есть несколько причин:

- ✦ тени делают страницу более привлекательной;
- ✦ тень создает иллюзию трехмерности объекта и всей страницы;
- ✦ тени легко создавать в графических редакторах.

Во избежание неприятностей при использовании теней, например, из-за конфликтующих источников света, придерживайтесь следующих рекомендаций:

- ✦ обращайте внимание на то, насколько светлой или темной получается тень, следите, чтобы яркость тени логически соответствовала расстоянию. Если тень темная, то объект кажется расположенным ближе к заднему плану; чем тень светлее, тем дальше объект зрительно отстоит от заднего плана и ближе к зрителю;

- ширина и высота тени также влияют на зрительное восприятие расстояния. Узкая тень приближает объект к заднему плану, а широкая или высокая отдаляют его;
- следите за тем, чтобы источник света был реалистичным. Если на странице есть два объекта, то тени от них должны падать в одну сторону.

Например, в первых двух перекрывающихся прямоугольниках, составляющих среднюю колонку в «Подсолнухе», тень падает от источника света, расположенного слева (рис. 5).

Ощущение света усиливается также за счет постепенного уменьшения яркости в заголовках правой колонки. Кажется, что свет угасает, и часть изображения пропадает из виду (рис. 6).

СОВЕТ

Если вы пользуетесь слоями в Photoshop, отметьте флажок Use Global Angle (Установить глобальный угол). Тогда угол, под которым падает свет, будет одинаков для всех изображений, с которыми вы работаете.

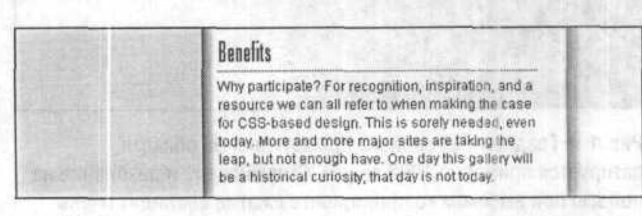


Рис. 5 ✦ Правильное использование источника света для создания согласованных теней

Рис. 6 ✦ Уменьшение освещенности по мере удаления от источника света добавляет реалистичности

О форме

Форма, как и цвет, оказывает на нас психологическое влияние. Дизайнер может задействовать определенные качества формы, чтобы вызвать у посетителей соответствующие эмоции эмоции.

Воздействие формы не ограничивается компьютерным или печатным дизайном. Люди пользуются формой еще с тех пор, как научились рисовать символы на песке или рассказывать истории с помощью рисунков на стенах пещер. Формы присутствуют повсеместно, они неотъемлемы от нашего мира да, кстати, и от всего, что делает человек. И при написании сложного математического уравнения, и для добавления архитектурных изысков к зданию и даже в религиозной и языческой символике (Святой крест, Звезда Давида, татуировки) – всюду формы являются частью нашей жизни и служат самовыражению.

В «Подсолнухе» самым ярким образом является цветок. Он притягивает взгляд не только своим цветом, но и формой.

Первичные формы

Окружности чаще всего ассоциируются с женским началом: теплота, уют, чувственность, любовь и другие понятия из того же ряда. Кроме того, окружности (или дуги) часто применяются для выражения идеи общности, целостности, стойкости, движения и безопасности. На многих сайтах, имеющих отношение к женщинам, общинам, соборной жизни и романтическим историям, встречаются окружности, которые усиливают эмоциональное восприятие.

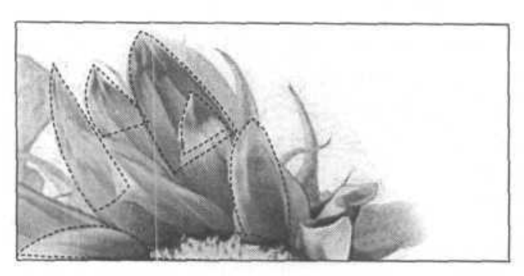


Рис. 7 ✦ При внимательном рассмотрении подсолнуха видны треугольные и круглые формы



Рис. 8 ✦ Прямоугольники помогают уравновесить дизайн

Треугольники считаются выражением мужского начала: силы, агрессии и динамики. У направления треугольника тоже есть свой смысл. Глаз неосознанно следует в направлении доминирующего угла. Если треугольник повернут вверх, он выражает движение и агрессию. Треугольник же, повернутый вниз, ассоциируется с пассивностью и бессилием.

Разумеется, с прямоугольниками (в том числе квадратами) тоже ассоциируются определенные представления: мощь и основательность (вероятно, из-за жесткости и предполагаемой массивности). Прямоугольные формы могут вызывать у зрителя ощущение упорядоченности, логичности, иерархичности и безопасности.

Комбинирование форм для достижения максимального воздействия

Внимательно приглядевшись к самому подсолнуху, мы обнаружим сочетание треугольников и окружностей. Конечно, они естественного происхождения, но Дарвас обработал изображение в Photoshop, подчеркнув цвет и форму цветка. В результате получилось очень чувственное изображение, сочетающее мужское и женское начала, и впечатление от него надолго запоминается (рис. 7).

В «Подсолнухе» строгие прямоугольники уравновешивают более чувственную природу цветка, что и формирует основу дизайна (рис. 8).

Различные формы можно комбинировать для оказания наибольшего воздействия на зрителя. Добавьте цвет и свет, и картина станет интереснее. А если еще правильно распорядиться пространством, то впечатление может быть очень ярким.

Пространства мне!

Наравне со светом, тенью и формой пространство также обладает интересными психологическими аспектами, которыми художники с давних времен пользовались для эмоционального воздействия на аудиторию.

Правильное использование пространства в визуальном дизайне позволяет достичь сразу нескольких целей: привлечь внимание к нужному месту страницы, удачно разместить текст и изображения, дать глазу на

чем-то отдохнуть. Это облегчает восприятие текста и оставляет место для обработки представленной информации.

Психологические и социальные аспекты

У пространства есть психологические и социальные аспекты:

- пространство и экономика взаимосвязаны. Конечно, вы слышали шутку о том, что женские юбки становятся короче, когда в экономике начинается спад. Имеющееся пространство и способ его использования значат очень много. Дизайн, загроможденный деталями, выглядит недостаточно стильно, тогда как хорошо сбалансированное размещение объектов в открытом пространстве несет более утонченный посыл;
- свободное пространство – это не просто отсутствие материальных предметов (на самом деле, оно усиливает впечатление от объектов). Обилие объектов вызывает чувство незащищенности и напряженности, тогда как наличие свободного места оставляет впечатление изысканности и легкости;
- у пространства есть культурные ассоциации. В разных культурах восприятие цвета и формы разнятся; также отличается и реакция на пространство. В некоторых обществах тесный физический контакт является предпочтительным способом общения, в других – к примеру, в Соединенных Штатах – комфортным считается общение на расстоянии.

Если применить все вышесказанное к дизайну, то станет ясно, что свободное пространство – это не просто то, что осталось на странице после размещения всех элементов, а неотъемлемая характеристика дизайна.

Позитивное и негативное пространство

Насколько важной составной частью дизайна является пространство, станет более понятно, если мы рассмотрим разницу между *позитивным* и *негативным* пространством.

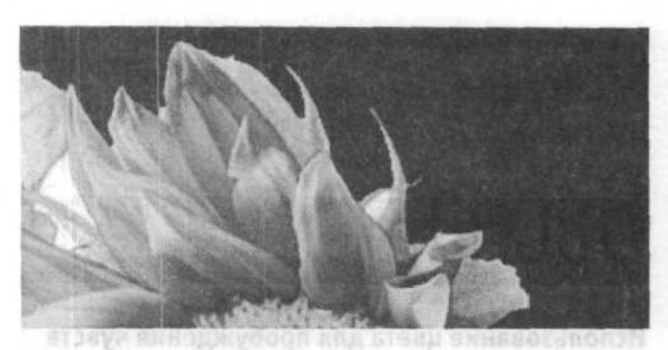


Рис. 9 ✦ Если зачернить свободное пространство, то станет видно, что оно определяет края объектов и само состоит из комбинации различных форм

Позитивным называется пространство, занятое объектами, составляющими элементы дизайна. *Негативное* пространство – это то, что свободно от объектов (фон). Однако полностью отделить позитивное пространство от негативного невозможно, они взаимозависимы и всегда присутствуют одновременно.

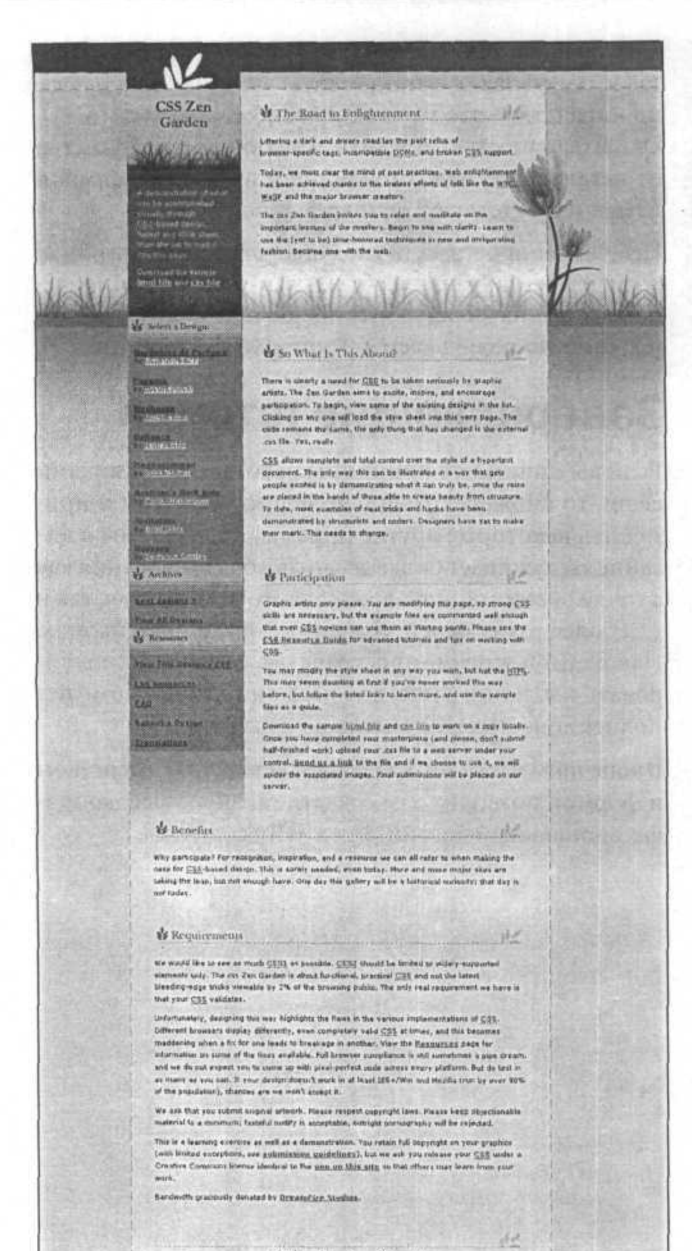
Негативное пространство помогает определить границы позитивного, а границы объектов, в свою очередь, определяют, где начинается негативное пространство, которое само по себе является некоторой формой (рис. 9).

Завершающие штрихи

Если вы понимаете принципы работы с источниками света, то сможете правильно нарисовать тени и применить некоторые другие приемы. Роль формы в дизайне заслуживает более детального исследования как с точки зрения общего зрительного восприятия, так и для более узких целей: создания логотипов и иконок. Наконец, ощущение пространства, умение манипулировать им, понимание, для чего оно нужно, поможет полнее донести свою мысль до зрителей.

В конечном итоге контроль над светом, пространством и формой позволит вам создать такой же высокопрофессиональный дизайн, как в «Подсолнухе».

3 Зак. 1942



Дизайнер **Боэр Аттила** (Boer Attila)
www.csszengarden.com/083

Весна

Использование цвета для пробуждения чувств

Как-то после холодной, скучной зимы Боэр Аттила проснулся и увидел за окном первый ясный, солнечный, весенний день, наполненный бодрящими, энергичными и радостными цветами. Этот переход от зимы к весне и вдохновил его на создание описываемого ниже дизайна, в особенности в плане выбора цветовой гаммы.

Давно известно, что цвет пробуждает в людях различные эмоции. Именно поэтому мы, например, называем красный цвет горячим. Красный ассоциируется с пламенем, со жгучим перцем. Эти идеи находят применение и в дизайне: скажем, цвета земли – коричневый и желтоватый – используются в упаковке для натуральных продуктов.

В «Весне» однотонные цвета зимы сталкиваются с цветами пробуждающейся весны. Получившуюся палитру не назовешь ни холодной, ни горячей, скорее она выражает теплоту и гармонию и создает у зрителя ощущение умиротворенности и уверенности.

Влияние цвета

Дизайнер, желающий донести свою мысль до аудитории, должен понимать, что восприятие цвета с учетом психологии, культурных особенностей и пола не менее важно, чем технические аспекты.

Ясный посыл

Конечно, контент и подбор графики ясно выражают смысл и назначение сайта. Но не менее важным средством передачи зрительной информации является цвет. Опытный дизайнер знает, какие цветы наилучшим образом подходят для передачи его идеи и соответствуют целевой аудитории. Цвет может быть так же – а иногда даже более – убедителен для человеческого глаза, как графические образы и текст.

Если ваша задача – создать сайт для группы, играющей металлический рок, то яркие цвета, например желтый и розовый на белом фоне, не смогут адекватно выразить идею. Металлисты ассоциируются с тьмой и, хотя яркие цвета, конечно, могут использоваться в дизайне, но лучше всего особенности этого музыкального жанра передаются темной, тяжелой цветовой гаммой.

Наоборот, если вы делаете сайт классического квартета, то очень темные цвета не смогут донести утонченность и легкость музыки. Выбор цветов, соответствующих теме и идее сайта, чрезвычайно важен, если вы хотите, чтобы дизайн решил поставленную задачу.

ПРИМЕЧАНИЕ

Иногда использование цветов, противоположных ожидаемым, может создать конфликт, и это тоже интересный подход к дизайну, приносящий подчас отличные результаты. Но чтобы отважиться на такое, нужно еще тщательнее планировать и тестировать сайт, чтобы выбранные цвета действительно достигли поставленной цели. Необходимо убедиться в наличии естественной корреляции между цветом и реакцией зрителя.

Цвет и человеческая психика

Воздействие цвета на человеческие эмоции – очень интересная тема. Между ними имеются сложные взаимосвязи, обусловленные многими факторами.

Люди с нарушенным цветовым восприятием, очевидно, испытывают несколько иные эмоции, чем люди с полноценным зрением. Примерно каждый двенадцатый американец страдает широко распространенной формой цветовой слепоты (неспособны отличить один цвет от другого, чаще всего зеленый от красного). Поэтому при выборе цветов следует иметь в виду демографический состав аудитории.

Оттенки цвета тоже играют роль. Эмоции, вызываемые ярко-зеленым и грязно-зеленым цветом, совершенно различны. Изменяет восприятие и добавление к цвету текстуры. В табл. 1 перечислены наиболее распространенные цвета и те понятия, с которыми они ассоциируются.

Таблица 1 ✦ Психологическое восприятие основных цветов

Цвет	Ассоциация
Красный	Мощь, энергия, любовь, страсть, агрессия, опасность
Синий	Доверие, консерватизм, защищенность, чистота, печаль, порядок
Зеленый	Природа, земля, здоровье, ревность, обновление
Оранжевый	Веселье, счастье
Желтый	Оптимизм, надежда, философское отношение к жизни, трусость
Фиолетовый	Величие, тайна, религия
Коричневый	Надежность, комфорт, выносливость, земля
Серый/серебристый	Интеллект, футуризм, скромность, грусть, упадок, элегантность
Черный	Сила, сексуальность, утонченность, тайна, страх, несчастье, смерть
Белый	Чистота, непорочность, точность, невинность, стерильность, смерть

Как видите, некоторые цвета воспринимаются неоднозначно, а иногда разные цвета вызывают одинаковые ассоциации, например черный и белый. Так, красный возбуждает страсть и одновременно является признаком опасности (а может быть, между этими эмоциями не так уж много различий?). Эти кажущиеся противоречия обусловлены множеством факторов, в том числе культурными и половыми различиями.

Для дизайна «Весна» выбраны нежные и приятные оттенки зеленого цвета, вызывающие воспоминание о траве и листьях. Легкий налет голубого, встречающийся в дизайне, тоже ассоциируется с природой (рис. 1).



Рис. 1 ✦ Палитра цветов дизайна «Весна» включает оттенки зеленого, намек на голубой и белый цвет

Цвет, культура и пол

Аттила по национальности венгр. Влияет ли культурная среда, в которой родился и живет художник, на выбор дизайна и восприятие цвета? Эксперты согласятся, что да, влияет. И точно так же впечатление, складывающееся у зрителя от созерцания «Весны», может зависеть от культуры и пола.

Особенности различных культур и пол обуславливают базовые ассоциации, обсуждавшиеся в предыдущем разделе. В табл. 2 приведена некоторая информация по этому поводу. Нетрудно предположить, что подходить к выбору цветовой гаммы нужно очень осторожно, особенно в такой глобальной среде, какой является Всемирная паутина.

ПРИМЕЧАНИЕ

Дополнительную информацию о цвете, а также ссылки на многочисленные статьи, книги и другие ресурсы по этой теме вы можете найти на сайте Colorcom профессора Дж. Л. Мортона (J. L. Morton) (www.colorcom.com). Еще один ценный источник информации о цвете – это сайт *Causes of Color* (http://webexhibits.org/causesofcolor).

ПРИМЕЧАНИЕ

Цветной вариант рис. 1 см. на вкладке.

Таблица 2 ✦ Влияние пола и культурной среды на восприятие цветов

Цвет	Ассоциация
Красный	В Китае это символ большой удачи. В сочетании с белым эта ассоциация усиливается. Многие женщины предпочитают красное синему
Синий	Во многих восточных странах это цвет бессмертия. У иудеев синий цвет символизирует святость, а в индуизме это цвет Кришны. Синий цвет почитают во всем мире, и повсюду он считается самым безопасным. Мужчины предпочитают синий красному
Зеленый	В США ассоциируется с деньгами, в других культурах это не так. В Ирландии зеленый вызывает сильные эмоции, поскольку ассоциируется с ирландскими католическими националистами. Женщины могут различить и назвать больше оттенков зеленого, чем мужчины
Оранжевый	В США по традиции оранжевая упаковка подчеркивает дешевизну товара. Выбирая такой цвет для сайтов коммерческих компаний, не следует забывать об этом. Для ирландцев в оранжевом скрыт глубокий эмоциональный подтекст, поскольку он ассоциируется с протестантизмом. Мужчины предпочитают оранжевое желтому
Желтый	Священный, императорский цвет в азиатских культурах. Женщины отдают желтому предпочтение перед оранжевым, поскольку он ассоциируется у них с теплотой и оптимизмом
Фиолетовый	В Европе – цвет траура. Также вызывает ассоциации с новыми и альтернативными религиями, поэтому его можно считать противоречивым. Фиолетовый редко встречается в природе, на ум приходят только немногие виды цветов и некоторые морские рыбы
Коричневый	Нейтральный цвет для всех культур и полов
Черный	Цвет траура и смерти в большинстве западных культур, да и многих других тоже
Белый	Цвет траура и смерти в большинстве азиатских культур. Олицетворяет чистоту и непорочность во многих западных обществах

ПРИМЕЧАНИЕ

Прежде чем окончательно определяться с дизайном, неплохо бы попробовать несколько палитр. В Adobe Photoshop сделать это легко (рис. 2). Создайте пустой документ и с помощью инструмента Color Picker поместите выбранные цвета на полоску, чтобы понять, как они будут выглядеть в сочетании. Затем можете подготовить тестовый документ и применить к нему эти цвета с помощью CSS.

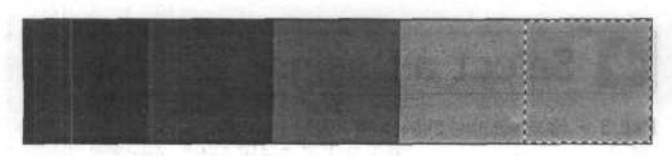


Рис. 2 ✦ Создание цветовой палитры в Photoshop

ПРИМЕЧАНИЕ

Цветной вариант рис. 2 см. на вкладке.

Цветовые палитры веб-сайтов

Выбирая цветовую палитру для своего проекта, не забывайте обо всем сказанном выше. Чтобы донести свою мысль до аудитории, надо понимать, как она отреагирует на те или иные цвета.

CSS может помочь на этапе создания прототипа сайта. Сделайте несколько таблиц стилей, по одной для каждой опробуемой палитры, и примените их к тестовому документу. Можно даже включить такую процедуру, как часть проверки пригодности сайта. В этом случае вы предъявляете фокус-группе различные цветовые схемы и наблюдаете за реакцией.

Изучение аудитории

Золотое правило общения – узнайте свою аудиторию. Поскольку человеческая психология, культурные влияния и пол накладывают отпечаток на восприятие цвета, необходимо ясно представлять, на кого будет рассчитан дизайн. Иногда ассоциации настолько универсальны, что применимы для любой группы. Сюда можно включить дизайн с обилием синего, коричневого и – как в случае «Весны» – зеленого цветов.

Эксперт по цвету Дж. Л. Мортон, чей сайт мы уже упоминали в этой главе, предлагает дизайнерам максимально придерживаться ассоциативныз связей. Если сайт рассчитан на детей, то выбирайте цвета, выражающие радость. Если вы хотите оформить сайт подчеркнуто элегантно, то в США прекрасно подойдут черный и серебристый цвета. Если вы обращаетесь к мужчинам, то, вообще говоря, изобилие синего не повредит. Если же сайт ориентирован на женщин, то красные и розовые тона предпочтительнее синих. Если сайт посвящен природе, то разумно брать зеленый, синий и коричневый цвета.

Цвета в CSS

К счастью, в CSS для работы с цветами есть много средств. К вашим услугам:

- **имена цветов**. В CSS определено 17 поименованных значений: black (черный), silver (серебристый), gray (серый), white (белый), purple (пурпурный), fuchsia (фуксия), green (зеленый), lime (лайм), olive (оливковый), yellow (желтый), navy (темно-синий), blue (синий), teal (бирюзовый), aqua (аквамарин) и orange (оранжевый). Пример:

```
color: orange;
```

- **RGB-значения (числовые)**. Можно использовать числовые значения для красной, зеленой и синей составляющих цвета, как в примере ниже – для светло-голубого. Узнать их можно в Photoshop (рис. 3) или любом другом графическом редакторе:

```
color: rgb(51,153,204);
```

- **RGB-значения (процентные)**. Можно задать процентные доли красного, зеленого и синего. В примере ниже получится фиолетовый цвет:

```
color: rgb(93%, 51%, 93%);
```

- **шестнадцатеричные значения**. Это шестизначное число в шестнадцатеричной системе счисления, которое можно получить с помощью Photoshop (рис. 4) или одной из множества других программ. Каждая пара символов в нем соответствует одной компоненте RGB. Такая запись применяется в Web уже очень давно, так что многие читатели с ней знакомы:

```
color: #0000FF;
```

Select a Design

Рис. 3 ✦ Нахождение значений RGB в Photoshop

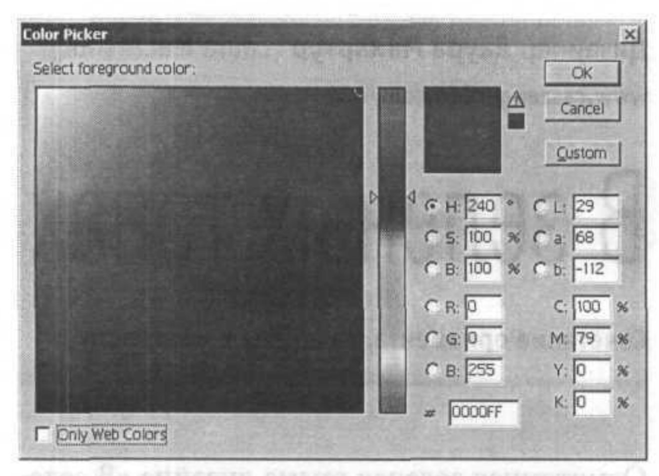


Рис. 4 ✦ Использование инструмента Color Picker в Photoshop для показа шестнадцатеричных значений цветов

ПРИМЕЧАНИЕ

Подробнее о системных цветах см. документ www.w3.org/TR/CSS21/ui.html#system-colors. Обращаем ваше внимание на то, что эти цвета не войдут в цветовой модуль, определенный в спецификации CSS 3.0.

ПРИМЕЧАНИЕ

Многие до сих продолжают пользоваться «безопасными» цветами. Но поскольку абсолютное большинство работающих сегодня компьютеров поддерживают миллионы цветов, то следовать этому соглашению уже необязательно, если только вы не рассчитываете на специфическую группу посетителей. Читатели, которые знают, какие 256 цветов считаются безопасными в Web, наверное, заметили, что цвета, использованные в дизайне «Весна», не входят в их число.

✦ **сокращенные шестнадцатеричные значения.** Такую нотацию можно применять лишь тогда, когда каждая пара состоит из двух одинаковых символов. Например, #FF6699 можно сократить, а #808080 – нельзя. Синий цвет из примера выше удовлетворяет указанному требованию, так что его шестнадцатеричное значение можно сокращенно записать в виде:

```
color: #00F;
```

✦ **системные цвета.** Это набор ключевых слов, определенных в CSS 2.1, которые позволяют пользоваться системными цветами, установленными на компьютере посетителя. Например:

```
color: WindowText;
```

Красочные выводы

Занимаетесь ли вы дизайном профессионально или для себя, начинать следует с ясного осознания того, что вы хотите сказать зрителю, и определения целевой аудитории. Зная это, вы скорее подберете цвета, усиливающие выразительность вашего послания.

«Весна» – это прекрасный пример цветового решения, точно соответствующего поставленным целям: выразить идею перехода от зимы к весне, вселить в зрителя чувство спокойного оптимизма и донести эти эмоции до глобальной аудитории, которая у сайта CSS Zen Garden, безусловно, имеется.

Дизайнер **Лаура Макартур** (Laura MacArthur)
www.csszengarden.com/022

В зеленых тонах

Сочетание орнамента, текстуры и контраста

Однотонная зеленая гамма дизайна «В зеленых тонах» в сочетании с пышной текстурой наводит на мысль о мшистой лесной подстилке и богатой бархатистой ткани. Лаура Макартур попыталась передать в этом дизайне чувство душевного равновесия и покоя.

Использованием повторяющегося узора и моделированием текстуры Макартур попыталась сбалансировать богатую деталировку с простотой остальных решений. Приглушенная однотонная цветовая гамма не загромождает дизайн, а плоские зеленые области оставляют место, на котором глаз может отдохнуть от узоров.

Орнамент



Рис. 1 ✦ Полутоновый орнамент крупным планом

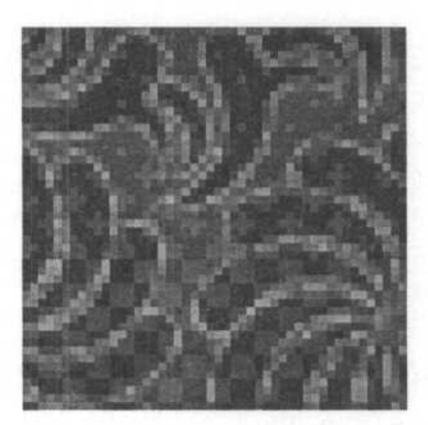


Рис. 2 ✦ Элемент цветочного орнамента крупным планом

Графика дизайна «В зеленых тонах» состоит в основном из повторяющихся орнаментов, которые используются повсюду. Есть два основных вида: один состоит из полутоновых точек, в другом прослеживаются цветочные мотивы (рис. 1 и 2). Орнаменты созданы путем сочетания элементов из галереи с фильтрами, имеющимися в Adobe Photoshop. Для большей зрелищности были разработаны некоторые дополнительные вариации, и то, что получилось, не выглядит скучным и повторяющимся.

Вообще говоря, орнамент – это повторяющийся визуальный элемент (или несколько элементов), заполняющий данную область. Орнаменты встречаются в самых разных видах графического дизайна, например одежды и интерьеров, а равно в природе: поры апельсиновой корки или повторяющиеся прожилки на листе.

Хороший орнамент – лучший друг веб-дизайнера. Обширные области, в которых нет или почти нет контента, нужно чем-то заполнить. В одних случаях для этого достаточно сплошной заливки цветом, в других имеет смысл оставить пустое белое место, а иногда нужна какая-то визуальная деталь сверх одного лишь цвета.

Ограничения, свойственные современному веб-дизайну, подталкивают к использованию орнаментов. Путем повторения простой небольшой картинки можно заполнить всю страницу, и в силу малого размера файла это обходится недорого в смысле затрат на передачу данных; повторяющимся изображением размером 500 байт можно, к примеру, замостить область шириной 900 пикселей. Управлять повторением позволяет определенное в CSS свойство background-color. Оно может принимать значения repeat, no-repeat, repeat-x и repeat-y, то есть повторять картинку, нарисовать только один раз или повторять в каком-то одном направлении.

```
h3 {
  background-repeat: no-repeat;
}
body {
  background: #669999 url(z_bgrnd.gif) repeat-x
  0px 1px;
}
```

Из этого примера видно, что свойство background-repeat можно задавать отдельно, независимо от других свойств фона, или вместе с другими значениями в составе стенографического свойства background.

Текстура

В Web не уделяется большого внимания тактильным ощущениям, впрочем, это характерно и для компьютерной индустрии в целом. Обратная осязательная связь часто встраивается в контроллеры игровых приставок, но помимо них почти не применяется в продукции, рассчитанной на массового потребителя.

В начале этого десятилетия компания Logitech экспериментировала с мышками под общим названием iFeel, которые позволяли чувствовать «текстуру» некоторых графических элементов операционной системы и веб-страниц. Например, когда курсор мыши проходил над гиперссылкой, ощущался слабый толчок. В последнее время нам не доводилось слышать об этой технологии, а Logitech таких мышек больше не выпускает. Хотя идея была интересной, никто так и не придумал, как ее применить, и дело заглохло.

Таким образом, текстуры в обычном понимании этого слова в Web отсутствуют; щелчок мышью – это, пожалуй, единственное тактильное ощущение. Но текстуру необязательно осязать, возможно и визуальное ее представление.

Скульптура может быть облачена в ниспадающие одежды, высеченные из камня. Мы знаем, что, коснувшись платья, ощутим лишь гладкость и твердость камня (рис. 3). Но в то же время каменное одеяние воспринимается как настоящая одежда, поскольку обладает теми же визуальными характеристиками.

Орнамент и текстура взаимосвязаны. Часто с помощью орнамента осознанно или неосознанно создается имитация текстуры. А многие текстурные элементы естественным образом формируют орнаменты за счет повторения схожих фрагментов.

Возможно, в основе орнаментов дизайна «В зеленых тонах» и не лежит фотография, но они безусловно напоминают реальные текстуры. Орнамент, составленный из точек, наводит на мысль о пупырчатой поверхности, а нечеткие квадратики походят на клетчатую ткань. Орнаменты с цветочными мотивами как будто сведены со старомодной подушки или богато украшенной драпировки. В целом при виде примененной структуры возникает образ удобного шезлонга, в который так и хочется опуститься.

Контрастность

Однотонная цветовая гамма в неумелых руках может оказаться опасной. Изобилие цвета и деталей приводит к проблемам с контрастностью, о которых забывать не следует.

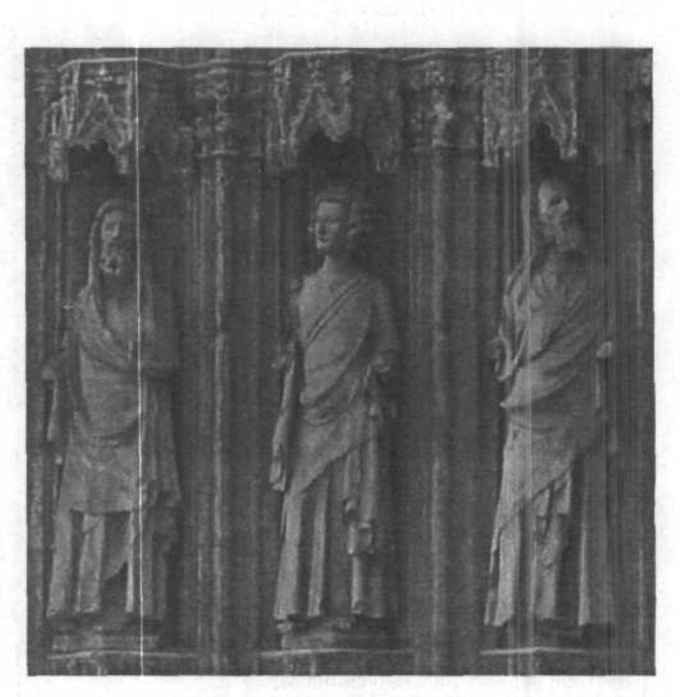


Рис. 3 ✦ На этих скульптурах из камня высечена текстура одежды

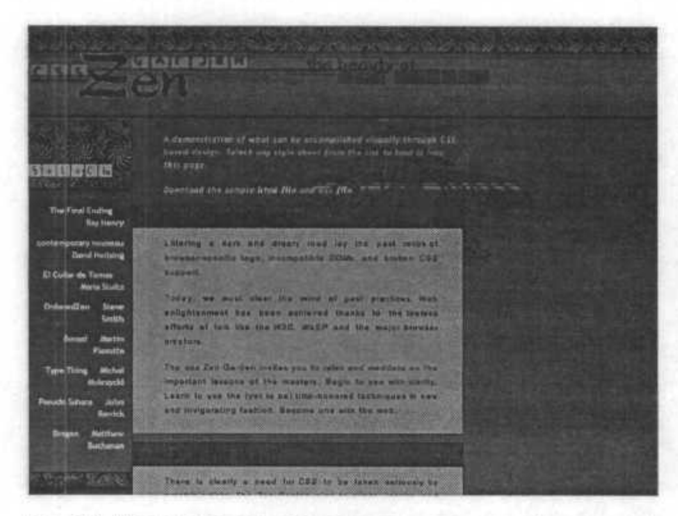


Рис. 4 ✦ Увеличение контрастности приводит к конфликту с другими деталями



Рис. 5 ✦ Одновременный контраст приводит к иллюзии мерцания на границах основных цветов

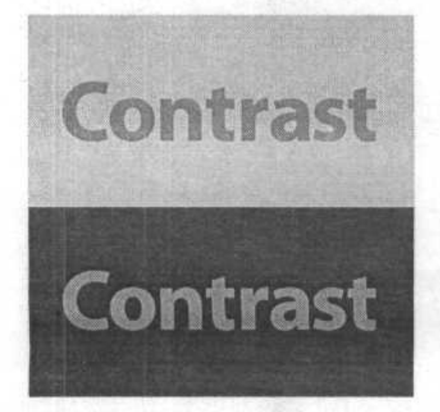


Рис. 6 ✦ Один и тот же цвет кажется более оранжевым на желтом фоне и более желтым на красном

ПРИМЕЧАНИЕ

Цветной вариант рис. 4–6 см. на вкладке.

Контрастность позволяет разграничить различные элементы. Если контрастность мала, то элементы зрительно смешиваются и текст становится нечитаемым. Если она слишком велика, то у зрителя возникает ощущение дискомфорта.

Высокая контрастность

Слишком высокая контрастность загромождает дизайн и ведет к «какофонии» утрированных визуальных образов. Узкий диапазон цветов в дизайне «В зеленых тонах» приемлем, потому что контрастность достигается за счет текстуры и типографских свойств текста. Вариант с более широкой цветовой гаммой был бы слишком кричащим (рис. 4).

На протяжении многих веков печатное слово наносилось черной краской на белую бумагу. Хотя цветные типографские краски изобретены уже давно, но черное на белом – это лучший из всех возможных контрастов, создающий наиболее комфортные условия для чтения. Вообще говоря, то же верно и для чтения с экрана: максимальная контрастность достигается при рисовании черного текста на белом фоне.

Впрочем, бывают и исключения. Поскольку компьютерный монитор в отличие от печатной страницы излучает свет, ему присущи специфические ограничения. Яркий ЖК-монитор в темной комнате может оказаться *слишком* ярким для комфортного чтения. Попытка прочитать с него что-нибудь в таких условиях может стать сущим мучением для глаз.

С точки зрения контрастности цвет представляет дополнительные проблемы. При некоторых сочетаниях близко расположенных цветов возникает оптическая иллюзия мерцания. Это так называемый *эффект одновременного контраста*. Когда два дополнительных цвета, например оранжевый и синий или зеленый и красный, находятся рядом, контраст получается особенно сильным и границы чрезмерно резки (рис. 5). Если такие дополнительные цветы перемежаются, то возникает неприятное для глаз напряжение.

Но в более тонких формах одновременный контраст может стать ценным инструментом для создания изысканных эффектов. Оранжево-желтый цвет может казаться более оранжевым на желтом фоне, но более желтым на оранжевом или красном фоне (рис. 6). Тусклые цвета на еще более тусклом фоне могут казаться более яркими: например, бледно-красный может приобрести насыщенность, будучи нанесен на серый фон.

Низкая контрастность

Слишком низкая контрастность создает общее впечатление тусклости дизайна. Контрастность необходима для того, чтобы отдельные элементы были различимы. В случае дизайна «В зеленых тонах» используются разные оттенки зеленого цвета – от светлых до темных; при меньшем разбросе цветов разрушилось бы впечатление от взаимодействия визуальных элементов, исчезло бы выделение и глазу было бы не на чем отдохнуть. В итоге визуальный баланс был бы нарушен (рис. 7).

Низкая контрастность часто применяется для создания эффекта утонченности. Пиксели – это относительно большие и тяжеловесные единицы для целей дизайна; невозможно провести линию тоньше одного пикселя, так как пиксель неделим. Но, уменьшив интенсивность отбрасываемой линией тени, можно создать иллюзию, будто линия стала тоньше (рис. 8).

Аналогично: путем уменьшения контраста между различными элементами дизайна можно добиться эффекта смягчения и уменьшения значимости. Например, менее важный текст можно было бы выводить более светлым цветом. А если нарисовать неактивную иконку или ссылку неконтрастными цветами, то пользователь получит визуальное указание на то, что она недоступна.

Возможности, которые предоставляет дизайнеру применение низкой контрастности, надо использовать правильно и с умом. Человек со стопроцентным зрением сможет без труда прочесть темно-серый текст на светло-сером фоне, но близорукому придется щуриться или увеличивать размер шрифта, текст черного цвета легче различим.

На разборчивость текста влияет и контрастность цветов. У разных цветов разные характеристики. Например, чистый желтый по природе своей светлее синего, поэтому желтый текст на белом фоне будет нелегко разобрать даже людям со стопроцентным зрением.

Цветовая слепота

Еще одним заслуживающим внимания аспектом контрастности является восприятие плохо сочетаемых цветов людьми, страдающими цветовой слепотой (рис. 9). Существуют две основных формы этой болезни и много разновидностей. Наиболее часто встречаются такие общие формы:

- ✦ **протаномалия/дейтераномалия**. При этом менее серьезном заболевании человек может видеть большинство цветов, но способность различать их понижена. Оно известно также под названием «аномальная трихромазия»;

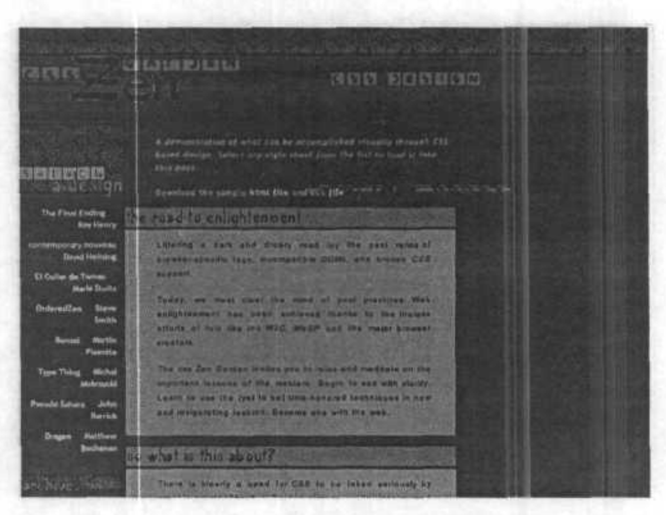


Рис. 7 ✦ «В зеленых тонах» при недостаточной контрастности

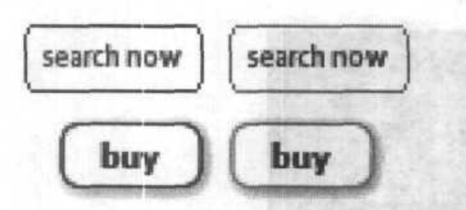


Рис. 8 ✦ Осветление внешних контуров кнопок заставляет их казаться тоньше

ПРИМЕЧАНИЕ

Цветной вариант рис. 7 см. на вкладке.





Рис. 9 ✦ Основной дизайн Zen Garden, пропущенный через фильтры: имитирующий протанопию (вверху) и имитирующий дейтеранопию (внизу)

ПРИМЕЧАНИЕ

Фильтр, позволяющий имитировать цветовую слепоту, имеется на странице Colorblind Web Page Filter (http://colorfilter.wickline.org).

ПРИМЕЧАНИЕ

Цветной вариант рис. 9 см. на вкладке.

✦ **протанопия/дейтеранопия**. Эта форма расстройства цветоощущения встречается реже, но более серьезна. Страдающие от нее не видят зеленого и красного цветов, оставаясь в желто-коричнево-синем мире. Болезнь называют также дихромазией.

У каждой формы заболевания есть разновидности, при которых понижена или полностью отсутствует способность различать комбинации красного, зеленого и реже – синего. Цветовой слепотой страдает примерно один человек из двенадцати, причем у мужчин она встречается гораздо чаще, чем у женщин.

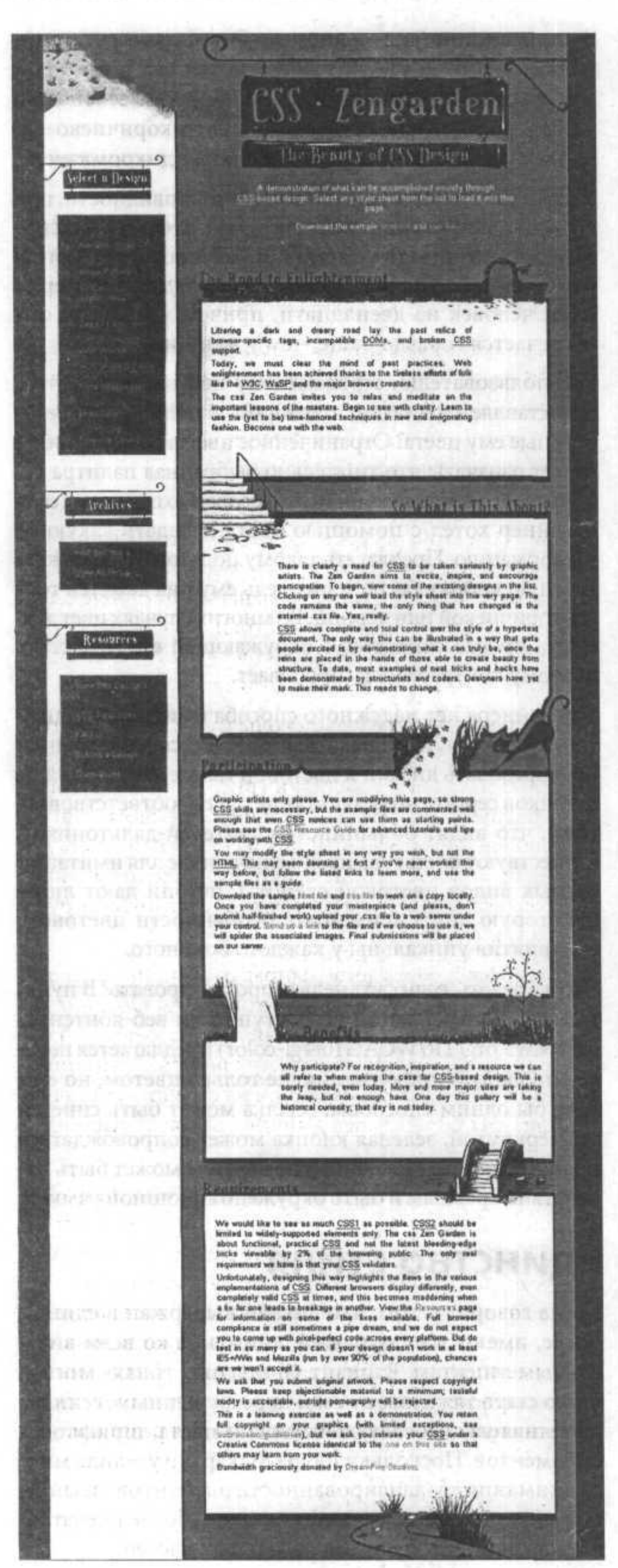
Как пользователю, страдающему цветовой слепотой, представляется дизайн, в котором используются недоступные ему цвета? Ограниченное цветовое восприятие может означать, что тщательно выбранная палитра искажается до неузнаваемости, что особенно печально, если дизайнер хотел с помощью цвета передать какую-то информацию. Предлагать такому пользователю нажать красную кнопку бесполезно, ведь ему она кажется темно-коричневой или черной. Во многих случаях цвет элемента просто сливается с окружающей его областью, поэтому контрастность пропадает.

У дизайнера нет надежного способа имитировать цветовую слепоту. На первый взгляд кажется достаточным преобразовать дизайн к цветовой гамме, состоящей из оттенков серого, но результат не будет соответствовать тому, что видят большинство зрителей-дальтоников. Существуют фильтры, предназначенные для имитации разных видов цветовой слепоты, но они дают лишь некоторую аппроксимацию, особенности цветового восприятия уникальны у каждого больного.

Что же делать, если сайт нельзя протестировать? В пункте 2 из «Рекомендаций по доступности веб-контента» (www.w3.org/TR/WCAG10/#gl-color) предлагается передавать любую информацию не только цветом, но еще хотя бы одним способом: ссылка может быть синей и подчеркнутой, зеленая кнопка может сопровождаться иконкой или текстом, предупреждение может быть напечатано красным и быть окружено сплошной рамкой.

Единство стиля

Когда говорят о том, что дизайн выдержан в едином стиле, имеют в виду равное внимание ко всем визуальным аспектам. Вариант «В зеленых тонах» мог бы легко стать тяжеловесным и перегруженным, если бы не тесная взаимосвязь текстуры и цвета, шрифтов и орнаментов. Поскольку Лаура Макартур уделила много внимания сбалансированности элементов дизайна, то окончательный результат оказался более впечатляющим, чем просто сумма отдельных частей.



Дизайнер **Шарлотта Ламберт** (Charlotte Lambert)
www.csszengarden.com/068

Баллада

Использование воображения для создания визуального потока и направления взгляда зрителя

Сама сущность сайта CSS Zen Garden располагает к спокойному созерцанию, и именно с этого начала иллюстратор Шарлотта Ламберт, приславшая дизайн «Баллада». Ламберт совершила мысленную прогулку по саду. А по пути ее воображению предстали различные образы: лестница, небольшой мостик, потягивающаяся кошка.

Выйдя из медитации и взявшись за кисть, она сделала несколько предварительных набросков, чтобы уточнить идею дизайна. Затем она запечатлела картины, возникшие в мозгу во время прогулки. Сначала они были нарисованы китайской тушью и белой краской на бумаге и даже на картонке из-под пива, а потом с помощью Photoshop были подготовлены изображения для Web.

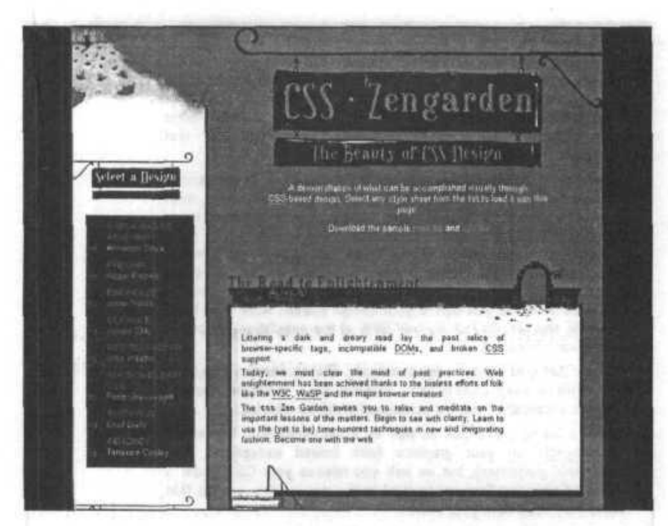


Рис. 1 ✦ Основной контент в «Балладе» отделен от навигационной и вспомогательной информации. Это один из способов привлечь внимание к содержанию и направить взгляд зрителя туда, куда хочет дизайнер

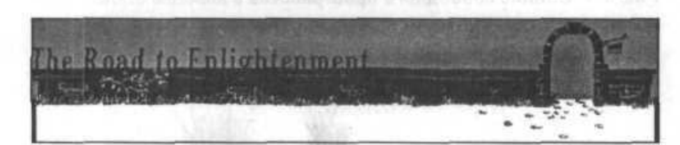


Рис. 2 ✦ Изображение калитки в «Балладе». Это рисунок очень важен для данного дизайна

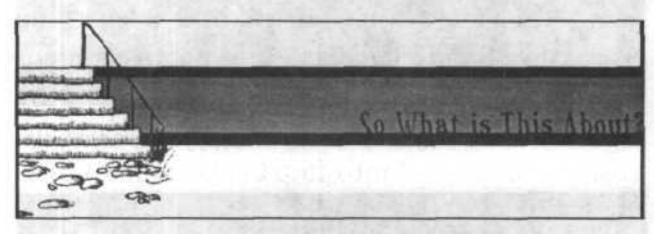


Рис. 3 ✦ От входа к лестнице – нас ведут по контенту как в метафорическом, так и в техническом смысле

Как направить взгляд зрителя

В хорошем дизайне взгляд перемещается в определенной последовательности от одной важной точки к другой, причем зритель даже не сознает, что его внимание намеренно привлекают к тем или иным местам на странице. Именно таким способом дизайнер передает свою идею, не становясь назойливым.

Чтобы направить взгляд, с помощью контента и графических образов создается визуальный поток, движущийся от одного важного места к другому. В «Балладе» мы наблюдаем отчетливое и вместе с тем ненавязчивое воплощение этой идеи.

Вынесение средств перемещения по сайту, ссылок на архивы и прочие ресурсы в левую часть отделяет навигационный контент от основного (рис. 1). Многие пользователи привыкли искать навигационные ссылки в левой или верхней части страницы, поэтому размещение их именно там позволяет посетителю сосредоточиться на главном содержании.

В дизайне Ламберт взгляд скользит от раздела к разделу, как если бы мы действительно прогуливались по саду. Мы начинаем с калитки (рис. 2), спускаемся по ступенькам (рис. 3) и идем дальше. У каждого участка сада свой неповторимый облик и достаточно места для контента. Заголовки описывают, что находится в каждом разделе, тем самым смысл контента интегрируется с визуальным потоком.

Еще одно важное средство направить взгляд – оставить места, на которых глаз может отдохнуть. Обычно это достигается за счет размещения пустых белых участков (рис. 4). С помощью цвета можно привлечь внимание читателя к ключевым местам на странице, а различные фигуры и линии задерживают взгляд и усиливают этот эффект.

В «Балладе» дизайнер решил поместить основной контент в блоки белого цвета, что создает хороший контраст для чтения и оставляет достаточно свободного места, на котором глаз может отдохнуть от напряжения. И хотя цветовая гамма исключительно проста и состоит только из коричневого, черного и белого цветов, она тем не менее интригует. Это зимний сад? Или осенний? Этот вопрос овладевает сознанием и заставляет искать дополнительные визуальные ключи для разгадки.

И, наконец, листья (рис. 5), вода, мостики, отпечатки лап (рис. 6) – все это необычные детали, за которыми следует взгляд зрителя, не устающего удивляться остроумию и прихотливости автора.

В терминах CSS

В последние годы появилось целое поколение дизайнеров новой школы, которые не мыслят в терминах традиционных для HTML табличных методов верстки. Вместо этого на первый план выходит применение CSS. Хотя все варианты дизайна на сайте CSS Zen Garden предполагают именно такой подход, но есть еще много дизайнеров старой школы, которые цепляются за слишком жесткие и зачастую более сложные методы верстки на основе таблиц. В «Балладе» автор выстроил макет сразу с помощью CSS. А что, если бы это было не так?

Беды старой школы

Если взглянуть на этот дизайн с точки зрения представителя старой школы, то быстро выясняется, что создание чего-то подобного с помощью таблиц выливается в кошмар. Возьмем всего два раздела из основной части страницы и попробуем построить для них макет в Macromedia Fireworks, пользуясь инструментом для разрезания картинок. Получившиеся кусочки нужно будет поместить в таблицу:

```
<table bgcolor="#ffffff" border="0"
cellpadding="0" cellspacing="0" width="597">
 <tr>
   <td rowspan="3"><img name="ballade_r1_c1"
   src="images/ballade_r1_c1.gif" width="30"
 height="887" border="0" alt=""></td>
   <td><img name="ballade_r1_c2" src="images/
 ballade_r1_c2.gif" width="537" height="273"
 border="0" alt=""></td>
   <td rowspan="3"><img name="ballade_r1_c3"
 src="images/ballade_r1_c3.gif" width="30"
 height="887" border="0" alt=""></td>
   <td><img src="images/spacer.gif" width="1"
 height="273" border="0" alt=""></td>
 </tr>
 <tr>
   <td><img name="ballade_r2_c2" src="images/
 ballade_r2_c2.gif" width="537" height="429"
 border="0" alt=""></td>
   <td><img src="images/spacer.gif" width="1"
 height="429" border="0" alt=""></td>
 </tr>
 <tr>
   <td><img name="ballade_r3_c2" src="images/
 ballade_r3_c2.gif" width="537" height="185"
 border="0" alt=""></td>
   <td><img src="images/spacer.gif" width="1"
 height="185" border="0" alt=""></td>
 </tr>
</table>
```



Рис. 4 ✦ Обилие свободного пространства в каждом блоке позволяет глазу отдохнуть и создает условия для комфортного чтения

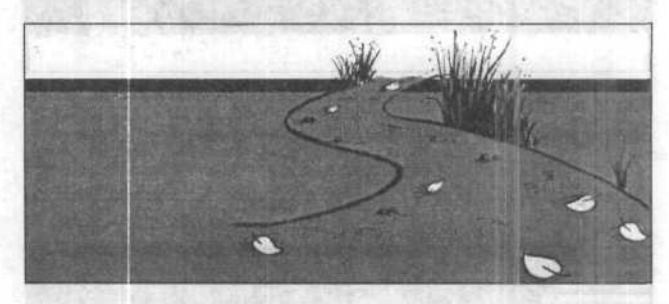


Рис. 5 ✦ Тропинка, ведущая к выходу из сада, усыпана листьями

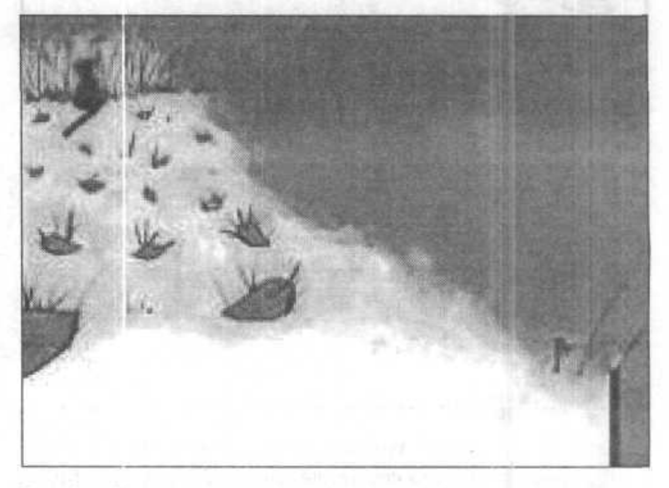
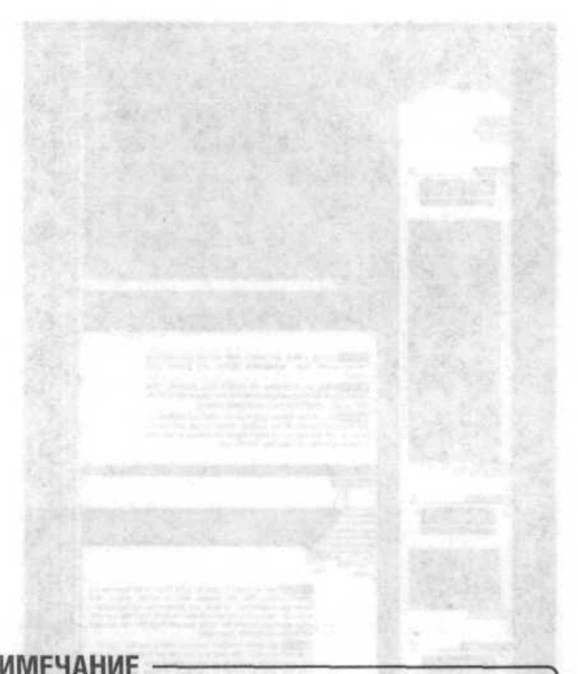


Рис. 6 ✦ Нарисованная кошка восседает в правом верхнем углу. «Баллада» изобилует приятными деталями, зрителю есть на что полюбоваться



Из этого кода видно, что макет усложняется в нескольких отношениях:

- при табличной верстке возникает много лишней разметки;
- применение пустых картинок в качестве разделителей вместо полей и отступов, имеющихся в CSS, без необходимости увеличивает «вес» страницы;
- большая часть картинок включена в саму разметку, что увеличивает размер файла за счет дополнительных тегов, а значит, и время загрузки;
- табличный дизайн по своей природе должен оставаться неизменным на всех страницах сайта. На каждой странице будет присутствовать разметка таблицы и все теги <img>, поэтому картинки и компоненты страницы перерисовываются при каждой загрузке. В случае CSS макет кэшируется, а для сайта CSS Zen Garden код, управляющий изображениями, загружается только один раз.

Конечно, все эти проблемы характерны для любых таблиц. Большинство дизайнеров понимают, что чрезмерное использование таблиц может быть причиной и других неприятностей, в том числе и в плане доступности, поскольку многие программы чтения с экрана, предназначенные для слепых, неправильно интерпретируют содержимое таблиц. А если учесть основной смысл данной работы – направление взгляда зрителя – то становится очевидным еще одно: неуклюжий старый подход разрушает впечатление непрерывности потока, поскольку сложный табличный макет обычно прорисовывается кусками.

ПРИМЕЧАНИЕ

Хотя макет на основе CSS обычно прорисовывается быстрее и более плавно, чем сложная табличная верстка, но известен так называемый феномен «появления нестилизованного контента». Он возникает в браузерах IE, когда CSS импортируется, а не присоединяется по ссылке. Если в заголовке документа присутствуют элементы `link` или `script`, то этот феномен не возникает. Во все варианты дизайна CSS Zen Garden включен элемент `script`, чтобы устранить эту проблему, поэтому прорисовка протекает гладко. Подробнее о феномене появления нестилизованного контента см. *http://www.bluerobot.com/web/css/fouc.asp*.

Подход новой школы

С помощью CSS гораздо проще создать дизайн, который будет загружаться и прорисовываться без рывков, приятно для глаза. Вместо сложных таблиц, нарезания изображений и применения графических разделителей дизайнер может сосредоточиться на стилизации отдельных элементов.

Мы вовсе не хотим сказать, что в дизайне на основе CSS вообще не должно быть встроенной графики. Конечно, это имеет смысл делать, когда графика привязана к контенту, например, если нужно поместить фотографию директора на странице «О компании» или снимок с экрана на сайте фирмы, производящей программное обеспечение. Суть в том, чтобы отделить изображения и другие элементы, служащие исключительно для презентационных целей, то есть являющиеся частью дизайна, а не самого контента.

ПРИМЕЧАНИЕ

Обертывать картинки заголовочными элементами полезно, когда графика является неотъемлемой частью заголовка. Применение такой техники подстановки изображений на сайте CSS Zen Garden обсуждается в главе 4.

Ниже показано, как графика, используемая в основной части страницы, задается с помощью фона заголовочных элементов:

```
#preamble h3 {
 background: url(road.jpg) top left no-repeat;
 height: 106px;
}
#explanation h3 {
 background: url(about.jpg) top left no-
repeat;
 height: 168px;
 }
 #participation h3 {
 background: url(particip.jpg) top left no-
repeat;
 height: 154px;
 width: 565px;
}
#benefits h3 {
 background: url(benef.jpg) top left no-
repeat;
 height: 171px;
}
#requirements h3 {
 background: url(req.jpg) top left no-repeat;
 height: 125px;
}
```

С помощью расширения Web Developer для браузеров Mozilla и Mozilla Firefox можно включить прорисовку контуров для блочных элементов и увидеть идентификаторы и классы всех элементов (рис. 7). Это очень удобно, когда нужно узнать точное расположение различных секций в дизайне.

Поместив контент внутрь элементов и применив на странице фоновую графику, дизайнер достиг сразу нескольких важных целей:

✦ за счет более простой разметки уменьшился вес страницы;

✦ вынесение большей части (или всего!) оформления в таблицу стилей способствует ускорению загрузки сайта, поскольку CSS-файлы кэшируются и браузеру не нужно повторно интерпретировать информацию о стилях при каждой загрузке страницы;

✦ сайт становится более доступным, так как в самом документе нет никаких презентационных элементов;

✦ сайтом становится проще управлять. Предположим, что с приходом весны мы хотели бы обновить «Балладу», включив зеленые и золотистые цвета вместо той понурой гаммы, в которой она нарисована сейчас. CSS позволяет справиться с этой задачей намного легче и с меньшими накладными расходами, так как мы заранее подумали о гибкости дизайна.

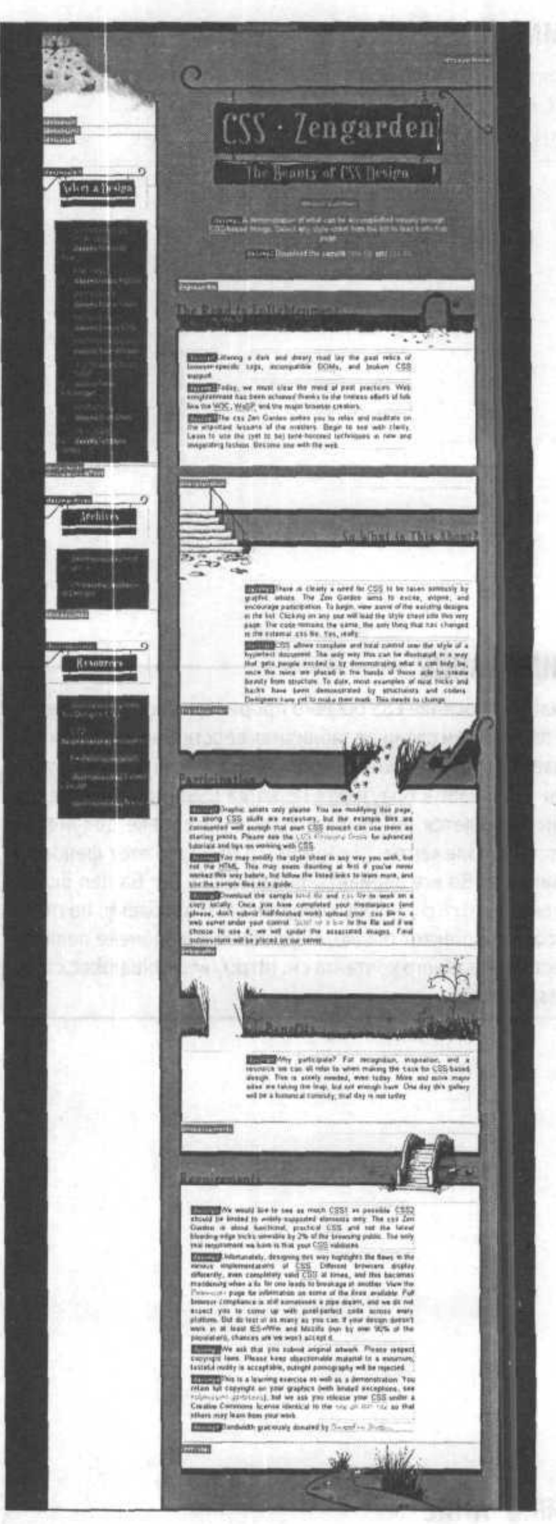


Рис. 7 ✦ Дизайн «Баллада», как его показывают различные полезные инструменты, имеющиеся в расширении Web Developer для Mozilla и Mozilla Firefox. Здесь для каждого блочного элемента нарисован контур, так что видно, каким образом дизайнер сверстал страницу и использовал фоновую графику

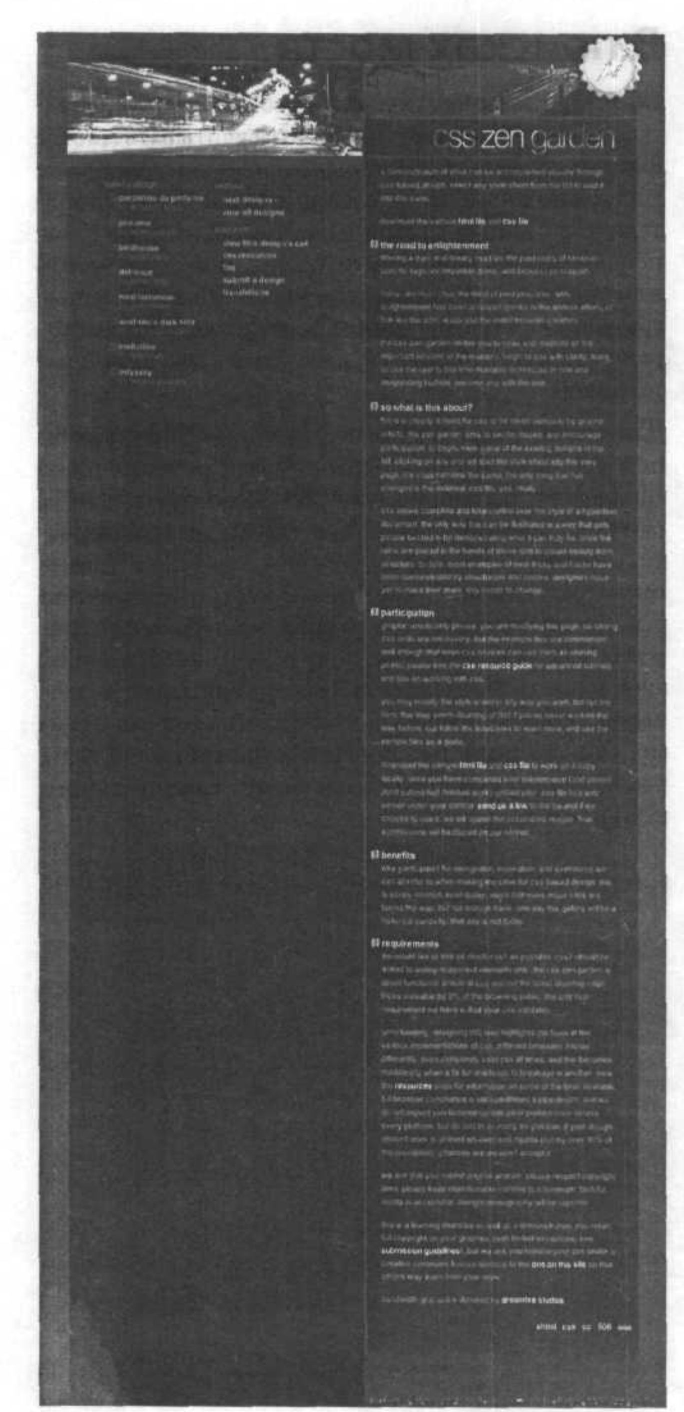
ПРИМЕЧАНИЕ

Загрузить расширение Web Developer для Mozilla и Mozilla Firefox можно со страницы http://chrispederick.com/work/firefox/webdeveloper.

В поисках моста

Эффективное проектирование веб-страницы – непростая задача. Дизайнеру приходится учитывать многие факторы. В «Балладе» наглядно продемонстрировано, как можно с помощью дизайна направить взгляд зрителя, с помощью цветов передать настроение и как использование CSS вместо таблиц может повысить качество сайта. Нам кажется, что дизайнер-новичок, сразу начавший с изучения CSS, будет свободен от ограничений и условностей, налагаемых традиционной техникой.

Для дизайнеров, которые переходят на CSS от табличной верстки, трудно вырабатывать новый способ мышления. Бесспорно, что CSS позволяет выйти за рамки многих ограничений, когда мы делаем наброски, работаем в Photoshop, готовим разметку и применяем к ней стили. Но вот найти мост, соединяющий разные технологии, – это проблема. Продемонстрировав удачное сочетание творческого воображения, живописи и CSS, Шарлотта Ламберт призывает дизайнеров пройти по мосту и взглянуть на мир по-новому. Она доказала, что новые прогрессивные методы работы могут освободить нас от пут традиционного веб-дизайна.



Дизайнер **Дэйв Ши** (Dave Shea)
www.csszengarden.com/064

Ночная поездка

Преобразование статической модели сайта в код и решение возникающих попутно задач

Вспоминается: поездка по городу вскоре после наступления сумерек, из динамиков звучит мелодия, исполняемая на саксофоне Джоном Колтрейном, за окном проносятся желтые огни, глубокая синева неба над головой. В «Ночной поездке» Дэйв Ши решил воссоздать настроение летнего вечера.

Темная палитра создает неяркий, теплый тон. Темно-оранжевый цвет текста и графики уравновешивается яркими оранжевыми пятнами, чтобы цветовая гамма не казалась излишне коричневой. Вдобавок местами в тексте и на фотографиях посверкивает синий цвет, чтобы не было ощущения монотонности, но контраст специально ослаблен и сделан спокойным.

Фотографии были выбраны из-за цвета и тематики: когда вы едете на машине после захода солнца, то ожидаете увидеть приглушенные огни, гладкую как зеркало поверхность воды и неясную синеву неба над горизонтом.



Рис. 1 ✦ Заголовок «Ночной поездки»

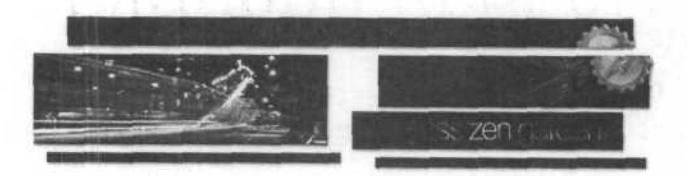


Рис. 2 ✦ Изображения, сохраненные отдельно от фона

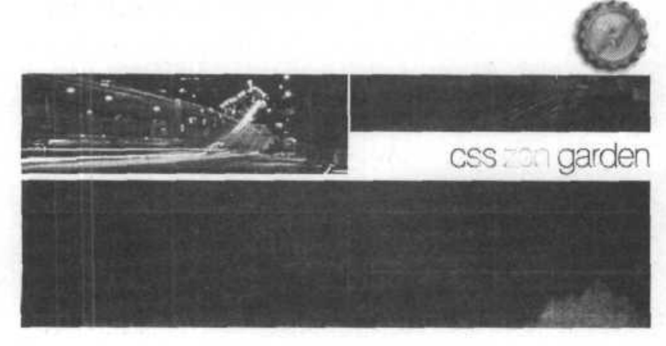


Рис. 3 ✦ Заголовок, разрезанный на куски

Визуализация кода

Когда Ши готовил дизайн «Ночной поездки» в Adobe Photoshop, он мало задумывался о том, как будет писать код. Сначала надо подготовить графику, а реализацию можно отложить на потом.

Но как превратить статическую модель в интерактивный сайт? Когда дизайн разбивается на куски, приходится сохранять разные изображения, которые позже предстоит собрать воедино с помощью CSS. Не существует четких правил, описывающих эту процедуру, ключом к успеху является гибкость. Если заранее иметь в виду альтернативы, то принять решение будет проще.

Альтернативные варианты

Рассмотрим заголовок «Ночной поездки» – набор состыкованных прямоугольников, на которые наложена яркая желтая эмблема (рис. 1). Здесь есть две проблемы: заполнить область фотографиями и линиями, а затем наложить поверх эмблему, задав для нее прозрачный фон. Поразмыслив, Ши остановился на двух возможных подходах:

✦ подготовить одну большую картинку, сохранив ее в формате JPG с очень малым коэффициентом сжатия, чтобы избежать искажения линий. В результате получился бы файл большого размера;

✦ разрезать картинку на куски, а затем собрать их воедино, привязав к различным элементам внутри `#pageHeader` и `#extraDivs` в конце страницы (куски, привязанные к последним элементам, можно поместить наверх с помощью абсолютного позиционирования) (рис. 2).

Ши выбрал второй способ, но только не стал превращать все прямоугольники в отдельные картинки, а поместил каждую из двух фотографий в JPG-файлы (рис. 3). Текст «css Zen Garden» был сохранен в формате GIF, а все остальное стало одним GIF-файлом с высоким коэффициентом сжатия. И вместо запутанного кода для сборки заголовка получилась ясное сочетание различных методов верстки с одновременным использованием GIF и JPG-файлов.

После того как файлы были сохранены, настало время подумать о том, как их собрать: какие картинки к каким элементам присоединить? Разумное использование атрибута background-image и техники подстановки изображений сделало ответ очевидным: поскольку внутри раздела #pageHeader элементов для этой цели недостаточно, следовало прибегнуть к элементам #extraDiv, находящимся в конце документа. В результате основная фоновая картинка была присоединена к элементу h1, а заголовок «css Zen Garden» – к элементу h2. Ниже код, относящийся к этим изображениям, выделен цветом:

```
h1 {
  width: 770px;
  height: 166px;
  margin: 0;
  background: #000 url(bg2.gif) top left no-
repeat;
}

h2 {
  width: 244px;
  height: 42px;
  background: #000 url(h1.gif) top left no-
repeat;
  position: absolute;
  top: 120px;
  margin: 0;
  margin-left: 495px;
}

h1 span, h2 span {
  display:none;
}
```

Дополнительные стили для span скрывают текст, вместо которого подставляются изображения, хотя этот метод сейчас считается устаревшим.

Магия абсолютного позиционирования

Итак, для размещения фотографий все готово, но при попытке включить их в заголовок возникает новая проблема: абсолютное позиционирование плохо сочетается с центрированием. Чтобы обойти эту трудность, Ши применил полезный трюк: он позиционировал родительские элементы вдоль левого края экрана, задав для них атрибут width равный 100%, чтобы растянуть на всю ширину окна браузера, а дочерние элементы – в данном случае span – центрировал в пределах родительских.

С учетом компенсации различных ошибок в браузерах получился такой код:

```
#extraDiv1 {
 position: absolute;
```

ПРИМЕЧАНИЕ

Методы подстановки изображений и связанные с ними проблемы обсуждаются на странице www.stopdesign.com/articles/replace_text.



> **ПРИМЕЧАНИЕ**
>
> Значение padding должно быть больше, чем вы ожидаете. Если задать для padding-right величину 100 px, то элемент сместится лишь на 50 px от центральной оси.



```
  top: 41px;
  left: 0;
  text-align: center;
  width: 100%;
 }
 #extraDiv1 span {
  background: transparent url(granville.jpg)
  top left no-repeat;
  display: block;
  margin-left: auto;
  margin-right: auto;
  height: 123px;
  width: 770px;
 }
```

Так как элемент центрирован, а добавление padding увеличивает ширину элемента, то отступ равномерно распределяется по обе стороны.

Это сработает, если дочерний элемент должен оказаться точно в центре, но если это не так? Задание для дочернего элемента свойства padding увеличивает его общую ширину, отодвигая тем самым от центра. Трюк здесь такой: отступ расширяет элемент и влево, и вправо, значит для достижения желаемого результата значение необходимо удвоить. Кроме того, отступ сдвигает элемент в направлении, противоположном тому, к которому он применен, следовательно, чтобы сдвинуть картинку влево от центра, надо задать свойство padding-right, и наоборот.

Ловкость рук

Разобравшись с заголовком, Ши столкнулся с новой проблемой при попытке увязать его с остальным контентом. Он хотел разделить колонки двойной вертикальной линией, но попытка применить к каждой колонке фоновое изображение, повторяющееся по вертикали, оказалась неудачной. Левая колонка (#linkList) не расширялась по вертикали на нужную величину, а правая в действительности состояла из трех разных элементов div (#quickSummary, #preamble и #supportingText).

Не имея единственного элемента, к которому можно было бы присоединить фоновую картинку, он присоединил ее ко всем трем, предположив, что они будут иметь одинаковую ширину и, значит, выровняются по горизонтали:

```
#quickSummary, #preamble, #supportingText {
  background: transparent url(bg4.gif) top left
  repeat-y;
  margin: 0 0 0 400px;
  width: 342px;
 }
```

Они и выравниваются, только возникает неожиданная сложность: между div образуется вертикальный разрыв (рис. 4). Но ведь поля ни для одного div явно не заданы, откуда же он берется?

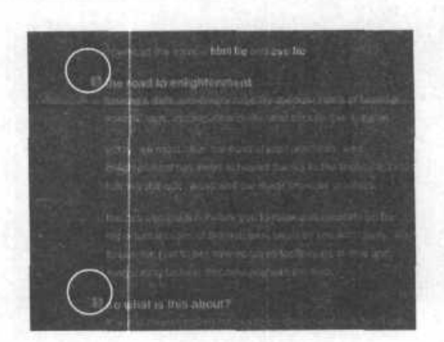


Рис. 4 ✦ Разрывы между границами

Схлопывающиеся поля

В модели *прямоугольных областей*, принятой в CSS, есть одна странность, получившая название схлопывающихся полей. Это означает, что если два элемента располагаются один над другим, причем для обоих заданы ненулевые поля и при этом нет ни отступов, ни границ, ни еще каких-либо явных условий, то полная высота поля будет меньше суммы двух полей. Так, поля шириной 20 пикселей не дадут в сумме промежуток шириной 40 пикселей, вместо этого поля схлопнутся, и вы получите все те же 20 пикселей.

> **ПРИМЕЧАНИЕ**
>
> Схлопывающиеся поля определены в спецификации CSS 2.1 следующим образом: «под схлопыванием полей понимается тот факт, что примыкающие друг к другу поля (без промежуточного контента, отступов и границ или иного разделяющего их промежутка) двух или более прямоугольных областей (соседних или вложенных) объединяются в одно поле». (*www.w3.org/TR/CSS21/ box.html#collapsing-margins*).

Но какое отношение это имеет к вышеупомянутой проблеме разрыва? Эффект схлопывания распространяется как на родительские, так и на дочерние элементы; поскольку для div не заданы вертикальные поля, то нижнее поле div схлопывается с нижним полем последнего дочернего абзаца. А так как полная величина поля отлична от нуля, то необходимо оставить заданное для абзаца поле шириной 10 px. Вот его-то и видно в нижней части объемлющего div (рис. 5).

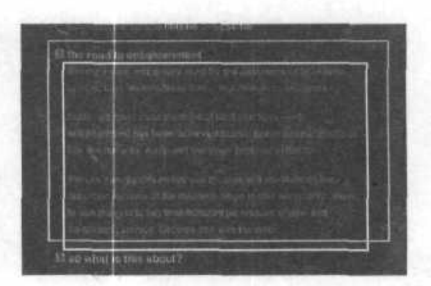


Рис. 5 ✦ Поля абзаца проступают сквозь поле объемлющего div

Коль скоро диагноз поставлен, проблему легко исправить. Если добавить к div небольшой вертикальный отступ в 1 px, то условия, необходимые для схлопывания полей, перестают удовлетворяться и разрыв исчезает:

```
#quickSummary, #preamble, #supportingText {
  background: transparent url(bg4.gif) top left
  repeat-y;
  margin: 0 0 0 400px;
  padding: 1px 0;
  width: 342px;
}
```

Фокусы с графикой

Определившись с заголовком и телом страницы, можно обратить внимание на эмблему в верхнем правом углу. Размещение картинки с прозрачными областями (в данном случае с тенью) поверх другой графики обычно означает, что при сохранении картинки надо учитывать, что же именно она перекрывает. Ши мог бы объединить слои в модели Photoshop и сохранить окончательное изображение эмблемы в формате GIF



Рис. 6 ✦ Эмблема, сохраненная вместе с фоном, прозрачности нет



Рис. 7 ✦ Эмблема, сохраненная без фона в виде GIF-файла с ограниченной прозрачностью



Рис. 8 ✦ Эмблема, сохраненная в виде прозрачного PNG-файла, наложенная на белый фон



Рис. 9 ✦ Эмблема, сохраненная в виде прозрачного PNG-файла, наложенная на фон с текстурой

или JPEG вместе с нижележащими фрагментами фотографии и линий, чтобы сберечь эффект прозрачности (рис. 6). А мог бы вместо этого удалить фоновые области и сохранить только прозрачный GIF, пожертвовав большей частью теней (рис. 7).

Но поскольку эмблема считается не столь важным графическим образом, Ши решил воспользоваться форматом PNG и избавить себя от некоторых сложностей. В формате PNG с 256 уровнями альфа-прозрачности эффект тени не пропадает, на каком бы фоне ни выводить изображение. На рис. 8 и 9 показано, как выглядит эмблема на двух разных фонах.

Ниже мы обсудим формат PNG более подробно, а пока отметим лишь, что браузер Microsoft Internet Explorer для Windows не понимает, что такое альфа-прозрачность, поэтому PNG-файл необходимо от него скрыть. Проще всего воспользоваться для этой цели селектором дочернего элемента, который этот браузер не поддерживает, и таким образом предлагать PNG-файл лишь тем браузерам, которые способны его нарисовать.

```
html>body #extradiv4 span {
 background: transparent url(seal.gif) top
right no-repeat;
}
```

Очевидно, в более формальном веб-проекте поддержка со стороны Internet Explorer даже для графики не первостепенной важности необходима. Хотя функциональность одноуровневой (да/нет) прозрачности в формате GIF более ограничена, но все же ее достаточно при надлежащем планировании. На сайте Zen Garden эксперименты поощряются, но у ваших клиентов может быть другое отношение.

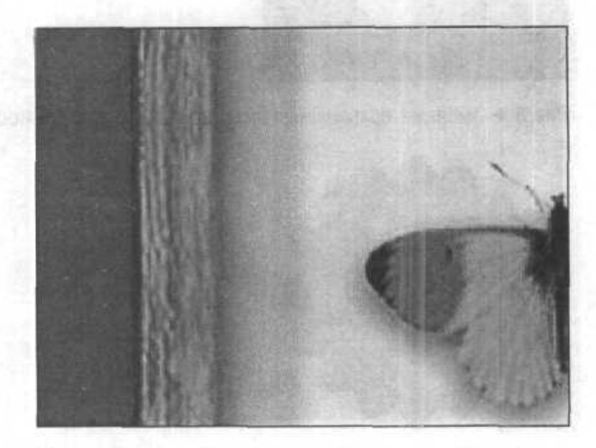
Один маленький шажок

Если в начале работы имеется лишь простая модель, а сроки поджимают, то, разработать полный макет на основе CSS может оказаться довольно трудно. Ключ к решению – разбить задачу на более мелкие; так, для верстки заголовка в «Ночной поездке» понадобилось два шага.

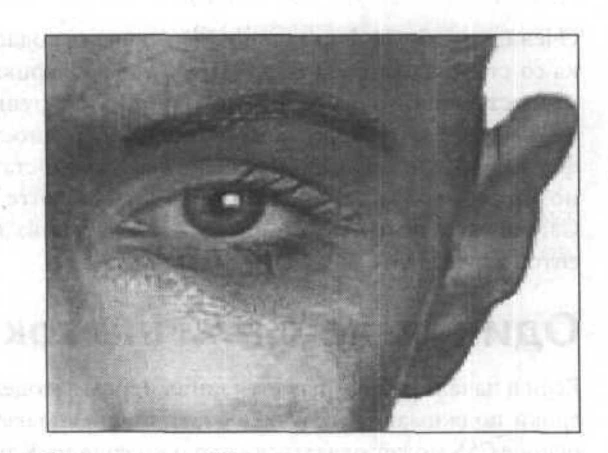
Если заранее знать о существовании разных возможностей, то, столкнувшись с трудностями, вы сможете принять более верное решение, будь то смена моделей позиционирования, регулировка полей или применение какого-либо эффекта к элементу.

SO WHAT IS THIS ABOUT?

Задний двор, с. 92



Энтомология, с. 98



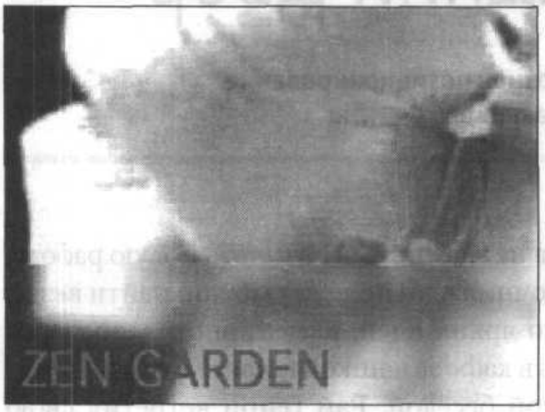
Прет-а-порте, с. 110



Монах, с. 116

3

Верстка



Белая лилия, с. 104

Не так уж бедно, с. 122

Для верстки страницы с помощью CSS необходимо сочетание технических знаний и творчества. Выбираете ли вы фиксированную или текучую верстку, выравниваете ли по левому краю или центру, в любом случае нужно четко понимать, как устроен макет в CSS.

В этой главе мы рассмотрим разные подходы к верстке, схемы позиционирования, применение плавающих элементов и научимся сочетать все это для получения полезного, привлекательного и – в некоторых случаях – неожиданного результата.

Дизайнер **Рэй Генри** (Ray Henry)

www.csszengarden.com/029

Задний двор

Абсолютное позиционирование и плавающие элементы

Иногда не знаешь, с чего начать новую работу, но источник вдохновения можно найти везде. Немало ярких идей, изменивших мир, зародилось в кафе за чашкой кофе с молоком и интересной беседой. Рэй Генри встретил свою музу в двух садах – Zen Garden и своем собственном. Так появился на свет «Задний двор». Цветовая гамма этого дизайна пришла с фотографии, на которой запечатлены цветы, растущие в саду на заднем дворе дома Генри.

Хотя дизайн с тремя колонками выглядит очень традиционно, у Генри во время работы возникло немало вопросов, и он поначалу не был уверен, что сумеет воспроизвести табличный макет с помощью CSS. Удастся ли поместить ссылки в правую колонку? Как это сделать? Может быть, для достижения нужного эффекта нужно верстать совсем по-другому? Но методом проб и ошибок Генри все-таки разрешил все проблемы поочередно.

CSS ZEN garden

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample **html file** and **css file**

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

PARTICIPATION

Graphic artists only please. You are modifying this page, so strong *CSS* skills are necessary, but the example files are commented well enough that even *CSS* novices can use them as starting points. Please see the **CSS Resource Guide** for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the *HTML*. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample **html file** and **css file** to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. **Send us a link** to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

BENEFITS

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

REQUIREMENTS

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical *CSS* and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your *CSS* validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid *CSS* at times, and this becomes maddening when a fix for one leads to breakage in another. View the **Resources** page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see **submission guidelines**), but we ask you release your *CSS* under a Creative Commons license identical to the **one on this site** so that others may learn from your work.

Bandwidth graciously donated by **DreamFire Studios**.

xhtml css cc 508 aaa

SELECT A DESIGN

contemporary nouveau by **David Hellsing**

El Collar de Tomas by **Hans Stoltz**

OrderedZen by **Steve Smith**

Bonsai by **Martin Plazotta**

Type Thing by **Michal Mokrzycki**

Pseudo Sahara by **John Barrick**

Dragen by **Matthew Buchanan**

Viscity by **Sooz**

ARCHIVES

next designs »

View All Designs

RESOURCES

View This Design's CSS

CSS Resources

FAQ

Submit a Design

Translations

ПРИМЕЧАНИЕ

Действительно, многоколонные схемы верстки в CSS явно не присутствуют, однако CSS-таблицы допускают и такой способ макетирования. Введены они в спецификации CSS2, но, к несчастью, Internet Explorer для Windows их не поддерживает, так что большого смысла в них пока нет. А жаль. Подробнее о CSS-таблицах см. документ по адресу www.w3.org/TR/CSS21/tables.html.

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Рис. 1 ✦ Элементы в нормальном потоке документа

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

The Road to Enlightenment

Рис. 2 ✦ При абсолютном позиционировании заголовок изымается из нормального потока документа, а на его месте оказывается следующий абзац

Основы макета

Чтобы сверстать страницу с несколькими колонками, необходимо кое-что знать о CSS. Не существует свойства «`left-column: 200px`», которое волшебным образом создало бы нужные колонки, поэтому приходится прибегать к различным трюкам.

Для создания макета, состоящего из колонок, есть два основных способа, и у каждого из них свои достоинства и недостатки. Из четырех схем позиционирования, имеющихся в CSS (absolute, static, relative и fixed), только абсолютное позиционирование позволяет изъять контент из нормального потока и поместить его в произвольное место. С другой стороны, хотя технически плавающие элементы нельзя назвать схемой позиционирования, но таким образом CSS позволяет зарезервировать место так, что остальное содержимое документа будет обтекать его.

Любой из этих методов обеспечивает контроль над элементами, достаточный для создания многоколонного макета. Плавающие элементы и абсолютное позиционирование можно использовать как вместе, так и порознь.

Абсолютное позиционирование

Из двух вышеупомянутых методов абсолютное позиционирование проще для понимания. Если размещать несколько элементов один за другим (это называется *нормальный поток документа*), как в HTML-разметке ниже, то получится линейная последовательность, изображенная на рис. 1. Этого и следовало ожидать.

```
<h1>Это заголовок</h1>
<p>Это текст</p>
<p>А это другой текст</p>
```

Применение к какому-либо элементу абсолютного позиционирования перемещает его (в данном случае заголовок) в указанное место, а оставшаяся «дырка» схлопывается (рис. 2). Абсолютно позиционированный элемент не оказывает влияния на своих соседей. Все выглядит так, как будто этого элемента вообще нет в документе (хотя мы его прекрасно видим). Синтаксически конструкция выглядит примерно так:

```
h3 { position: absolute; top: 100 px; }
```

Достоинство метода абсолютного позиционирования в том, что он обеспечивает точный контроль над положением данного элемента. Но раз элемент изъят из нормального потока, то он не может воздействовать на остальные элементы. Иногда это становится проблемой, как мы покажем в следующем размере.

Плавающие элементы

С плавающими элементами разобраться сложнее, поэтому мы прибегнем к помощи рисунков. На рис. 3 показан длинный фрагмент текста; если поместить в него фотографию, то она, естественно, займет какое-то место, а остаток текста сдвинется вниз (рис. 4). Чтобы избежать разрыва, нужно, чтобы текст заполнил свободное место справа от фотографии (рис. 5).

Именно в этом и состоит суть плавающих элементов в CSS: фотография вставляется в текст и делается плавающей, а текст обтекает ее, чтобы не было уродливого разрыва.

Синтаксис прост, но овладеть им нелегко. Фрагмент таблицы стилей, описывающий только что рассмотренный пример, мог бы выглядеть примерно так:

```
#photo { float: left; margin: 10px 10px 10px 0; }
```

Достоинство верстки методом плавающих элементов состоит в том, что элемент не изымается из нормального потока и соседние элементы знают о его положении. Но из-за различий в поддержке со стороны браузеров плавающие элементы тоже не всегда идеальны для макетирования, и для того чтобы выбрать один из двух методов, необходимы некоторые практические познания.

Методы верстки

Разметка в примере ниже содержит базовые блоки для организации трех колонок и хвостовика. Главная область контента располагается в коде раньше обеих боковых колонок. Это сделано для удобства поисковых машин и ради пользователей, которые при просмотре страниц отключают поддержку CSS: чем раньше текст появляется в HTML-коде страницы, тем меньше элементов навигации и прочих ссылок придется пропустить, чтобы добраться до важного контента.

```
<div id="wrapper">
 <div id="contentArea">
  <p>(здесь заголовки и текст)</p>
 </div>
 <div id="leftPanel">
  <p>(здесь ссылки и различные текстовые
элементы)</p>
```

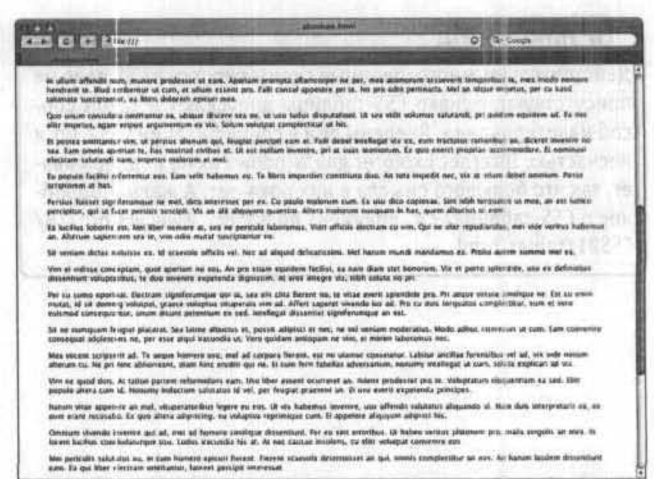


Рис. 3 ✦ Сплошной текст

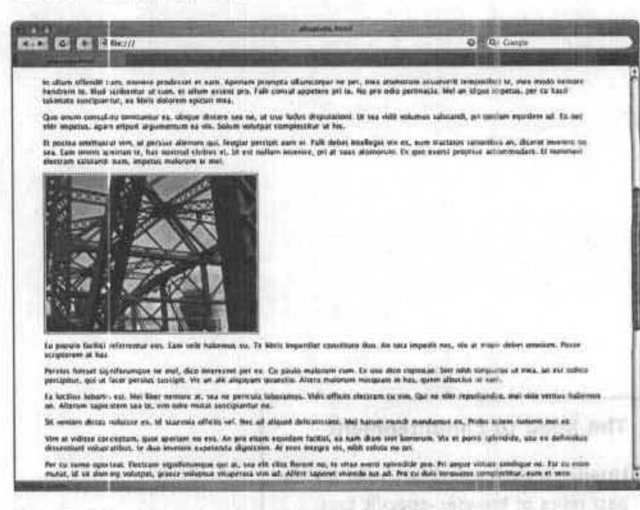


Рис. 4 ✦ После добавления фотографии в тексте образовался разрыв



Рис. 5 ✦ Текст обтекает фотографию и заполняет разрыв

> **СОВЕТ**
>
> Internet Explorer для Windows почему-то иногда удваивает поля для плавающих элементов. Если значения полей заметно больше 10 пикселей, то вы будете долго чесать в затылке и чертыхаться. Чтобы обойти эту проблему, задайте для плавающего элемента свойство `display:inline`. Подробнее об этой ошибке можно прочитать на сайте www.positioniseverything.net/explorer/doubled-margin.html.



Рис. 6 ✦ Простой макет с тремя колонками



Рис. 7 ✦ Тот же макет, но выровненный по центру

```
 </div>
 <div id=»rightPanel»>
  <p>(здесь ссылки и различные текстовые
элементы)</p>
 </div>
 <div id=»footer»>
  <p>(Авторское право, ссылки и т.д.)</p>
 </div>
</div>
```

Чтобы по-быстрому сверстать макет, в котором используется абсолютное позиционирование, не нужно долгого планирования. Задайте для всех трех колонок фиксированную ширину, позиционируйте каждую из них, ведя отсчет от левого верхнего угла документа, и временно забудьте о хвостовике. Получится так, как это изображено на рис. 6.

```
#contentArea {width: 450px; position:
absolute; top: 0; left: 150px; }
#leftPanel {width: 150px; position: absolute;
top: 0; left: 0 }
#rightPanel {width: 150px; position: absolute;
top: 0; left: 600px; }
#footer {display: none; }
```

Сейчас все прижато к левому краю, а как можно отцентрировать (рис. 7)? Легко. Модель позиционирования обладает полезной особенностью: если вы абсолютно позиционируете элементы, содержащиеся в другом элементе, который тоже абсолютно позиционирован, то значения `top` и `left` отсчитываются не от положения корневого элемента `html` (то есть от левого верхнего угла окна браузера), а от левого верхнего угла объемлющего элемента.

Непонятно? Объемлющий элемент ведет себя как новая точка отсчета для абсолютного позиционирования всего, что в нем находится. Поэтому если элемент `#wrapper` позиционировать и центрировать, то все колонки останутся внутри него. Приведенная ниже таблица стилей показывает, как связать все это воедино (свойство `text-align` включено для того, чтобы обойти «глюк» в Internet Explorer 5 для Windows, который не центрирует элементы со свойством `margin`, равным `auto`).

```
body {
 text-align: center;
}
#wrapper {

 width: 750px;
 margin: 0 auto;
 text-align: left;
 position: relative;
}
#contentArea {
 width: 450px;
 position: absolute;
```

```
 top: 0;
 left: 150px;
}
#leftPanel {
 width: 150px;
 position: absolute;
 top: 0;
 left: 0
}
#rightPanel {
 width: 150px;
 position: absolute;
 top: 0;
 left: 600px;
}
```

Но что делать, если нужен хвостовик? Поскольку абсолютно позиционированные колонки не влияют на соседние элементы, то хвостовик их игнорирует и располагается так, как если бы этих колонок не существовало. В данном случае он окажется первым элементом на странице.

Чтобы решить эту проблему с помощью абсолютного позиционирования, нужно заранее знать высоту каждой колонки и задать позицию хвостовика так, чтобы он оказался в самом низу. Не обращаясь к JavaScript, сделать это невозможно, остается только перейти от абсолютного позиционирования к плавающим элементам.

На рис. 8 хвостовик автоматически позиционируется под колонками благодаря использованию плавающих элементов.

```
body {
 text-align: center;
}
#wrapper {
 width: 750px;
 margin: 0 auto;

 text-align: left;
 position: relative;
}
#contentArea {
 width: 450px;
 float: left;
}
#leftPanel {
 width: 150px;
 float: left;
}
#rightPanel {
 width: 150px;
 float: left;
}
#footer {
 clear: both;
}
```

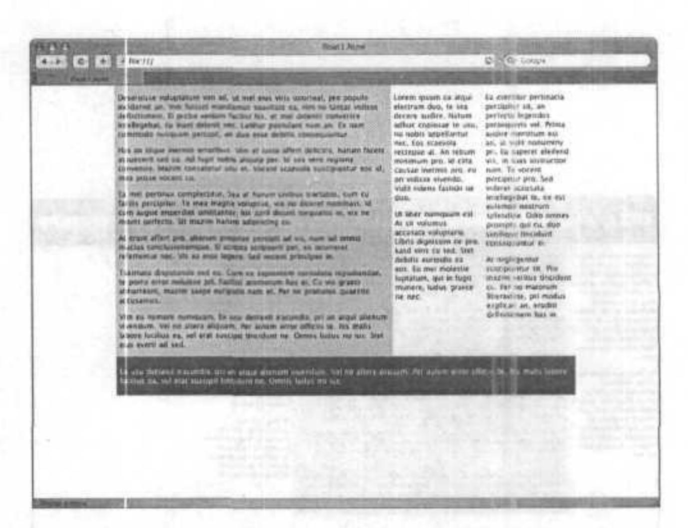


Рис. 8 ✦ Тот же макет, но с плавающими элементами и хвостовиком

СОВЕТ

Дополнительные сведения о решении на основе отрицательных полей см. на демонстрационной странице Райана Брилла (Ryan Brill) www.alistapart.com/ articles/negativemargins.

Но теперь левая и средняя колонки поменялись местами! Поскольку плавающие элементы остаются в нормальном потоке документа, их порядок в исходной HTML-разметке гораздо важнее, чем при абсолютном позиционировании. Самый простой способ решить эту проблему – изменить порядок следования левой и средней колонок в HTML-коде. Есть и другие подходы, связанные с использованием отрицательных значений полей. Но в девяти случаях из десяти достаточно поменять порядок элементов.

Все абсолютно сочетаемо

Как и многие другие, приступающие к изучению верстки на основе CSS, Рэй Генри для своего макета с тремя колонками выбрал абсолютное позиционирование. Во многих случаях и уж, конечно, для «Заднего двора» это вполне здравое решение, поскольку методика эффективна и проста для понимания.

Но не для всех сценариев это решение идеально, иногда возникают сложности. Если необходим хвостовик или элементы на странице должны что-то знать о взаимном расположении, на помощь приходят плавающие объекты. Само по себе ни одно решение не идеально, но они неплохо дополняют друг друга. Пользуйтесь и тем и другим, в сочетании они способны реализовать все, что вы только можете себе вообразить.

4 Зак. 1942

Дизайнер **Джон Хикс** (John Hicks)
www.csszengarden.com/030

Энтомология

Центрирование в современном веб-дизайне

Интересно, откуда берутся идеи? В случае «Энтомологии» идею дизайна навеял музыкальный клип – сингл «Булавка», исполняемый Yeah Yeah Yeah. Джон Хикс хотел создать визуальное воплощение концепции сада, которое тем не менее садом не являлось бы. Видеоряд в «Булавке» содержит изображения наколотых бабочек, которые идеально отвечали замыслу Хикса.

Нарисовав грубый эскиз макета, Хикс сразу перешел к CSS, не создавая предварительно модель в Adobe Photoshop. Поскольку это был его личный проект, а у него хватало и другой работы, то он не стал тратить время на все шаги стандартной процедуры дизайна. Он рассказывает, что даже надел тропический шлем и положил рядом сачок, чтобы создать реалистичную обстановку! То, что получилось в результате, резко отличается от присущей Хиксу манеры.





(select a design:)
CONTEMPORARY NOUVEAU by David Hellsing
EL COLLAR DE TOMAS by Maria Stultz
ORDEREDZEN by Steve Smith
BONSAI by Martin Plazotta
TYPE THING by Michal Mokrzycki
PSEUDO SAHARA by John Barrick
DRAGEN by Matthew Buchanan
VIVACITY by Soso



(archives:)
next designs »
View All Designs



(resources:)
View This Design's CSS
CSS Resources
FAQ
Submit a Design
Translations

A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample *html file* and *css file*

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden *invites you to relax and meditate* on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. *The only way* this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

PARTICIPATION

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample *html file* and *css file* to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. *Send us a link* to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

BENEFITS

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

REQUIREMENTS

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the Resources page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You *retain full copyright on your graphics* (with limited exceptions, see *submission guidelines*), but we ask you release your CSS under a Creative Commons license identical to the *one on this site* so that others may learn from your work.

Bandwidth graciously donated by *DreamFire Studios*. Coming soon: *Zen Garden, the book*.

{xhtml} {css} {cc} {508} {aaa}



Рис. 1 ✦ Верстка с фиксированной шириной



Рис. 2 ✦ Текучая верстка

Фиксированная и текучая верстка

Макеты для Web традиционно принято относить к одному из двух видов: *фиксированные* и *текучие*. При фиксированной верстке задается конкретная ширина. Контент можно выровнять по левому краю, правому краю или центру, но он не должен выступать за пределы области указанной ширины. Примером может служить дизайн «Энтомологии» (рис. 1).

Текучий, или *динамический* (liquid, fluid, dynamic), дизайн отличается тем, что ширина задается в процентах, а не в абсолютных величинах, так что макет «растекается», стремясь заполнить всю ширину окна (рис. 2).

Возможны и комбинации этих подходов. Взять, к примеру, трехколонный макет, в котором для боковых колонок задана фиксированная ширина, а для средней – процентное отношение (или ширина не задана вовсе). В этом случае контент займет все доступное пространство, но средняя колонка будет сжиматься или расширяться при изменении размеров окна браузера или разрешающей способности монитора.

Как и во многих других случаях, пристрастия и мода постоянно меняются. В самом начале развития Web предпочтение отдавалось фиксированной верстке. Но со временем стали обсуждаться достоинства текучего дизайна, а спустя несколько лет он стал очень популярен.

В последние два года вновь растет популярность фиксированной верстки, особенно среди дизайнеров, приверженных стандартам. Когда мы работали над книгой, почти все ведущие специалисты в этой области применяли фиксированный дизайн как на собственных сайтах, так и при работе на заказ. Это интересная тенденция – надо бы разобраться, почему фиксированная верстка может нравиться дизайнерам больше, чем текучая.

Прежде всего, фиксированная верстка оставляет дизайнеру больше контроля над пространством, в котором он может работать. Проще управлять длиной текстовых строк, а также взаимным расположением и близостью текста и изображений. Поэтому весьма вероятно, что фиксированная верстка вновь выйдет на первый план.

У текучего дизайна тоже есть свои преимущества. Если заполнено все окно браузера, то сама собой решается проблема зрительного баланса, возникающая в случае, когда фиксированная, прижатая к левому краю верстка занимает лишь часть доступного пространства. Но за это приходится платить потерей контроля над длиной строк, потоком и взаимным расположением элементов. Для опытных дизайнеров это серьезные проблемы, а вариантов их решения совсем немного.

Чтобы одновременно иметь возможность контроля, как при фиксированной верстке, и решить проблему зрительного баланса, дизайнеры часто центрируют контент. Тогда свободное пространство равномерно распределяется по обе стороны занятой области, что позволяет избежать широкого пустого места справа.

Способы центрирования

Центрирование с помощью CSS оказывается не совсем тривиальным делом, так как узаконенный способ горизонтального размещения по центру работает неодинаково в разных браузерах. Хикс столкнулся с этой проблемой, работая над «Энтомологией», и решил ее обходным путем. Но прежде чем знакомиться с его подходом, рассмотрим стандартные методы центрирования макета с помощью таблиц стилей.

Центрирование с помощью автоматически устанавливаемых полей

Предпочтительный способ центрирования любого элемента по горизонтали – это использование свойства `margin`, значения `left` и `right` которого задаются равными `auto`. Чтобы этот способ сработал, необходимо создать объемлющий `div`. При этом обязательно должна быть задана его ширина.

```
div#container {
  margin-left: auto;
  margin-right: auto;
  width: 168px;
}
```

На рис. 3 приведен пример использования такой техники для одной из бабочек в дизайне Хикса. Картинка помещена в отдельный `div` и центрирована с помощью автоматических полей.

Этот прием прекрасно работает почти во всех современных браузерах, даже в Microsoft Internet Explorer 6 для Windows при условии, что установлен режим

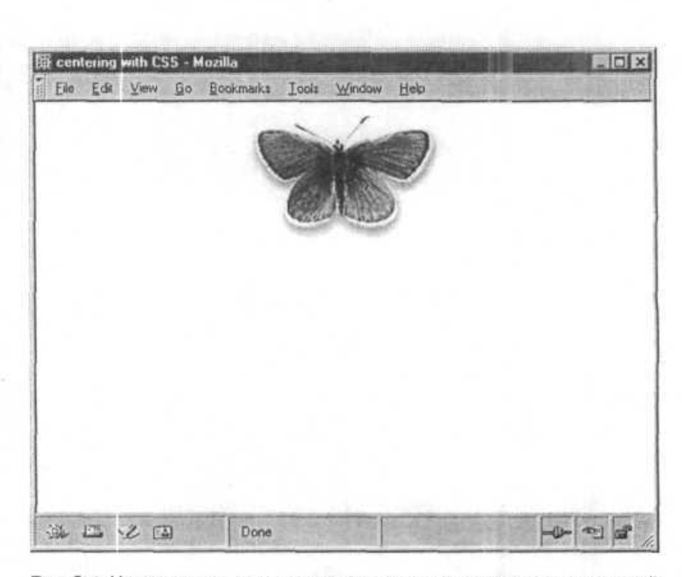


Рис. 3 ✦ Центрирование элемента с помощью автоматических полей

ПРИМЕЧАНИЕ

Режим совместимости – это часть технологии производства браузеров, которую часто еще называют переключением DOCTYPE. В XHTML-коде сайта CSS Zen Garden используется подходящий для активации режима совместимости DOCTYPE. Более подробную информацию на эту тему можно найти в статье Матиаса Гутфельдта (Matthias Gutfeldt) «Doctype switching and standards compliance: An overview» на странице http://gutfeldt.ch/matthias/articles/doctypeswitch.html.

совместимости. К сожалению, в более ранних версиях Internet Explorer для Windows он не работает, поэтому его применимость ограничена.

В табл. 1 приведены сведения о поддержке этой техники в разных браузерах.

Таблица 1 ✦ Поддержка автоматических полей в разных браузерах

Браузер	Версия	Поддержка
Internet Explorer	6.0, режим совместимости	Да
Internet Explorer	6.0, особый режим	Нет
Internet Explorer	5.5 Windows	Нет
Internet Explorer	5.0 Windows	Нет
Internet Explorer	5.2 Macintosh	Да
Mozilla	Все текущие версии	Да
Mozilla Firefox	Все версии	Да
Netscape	4.x	Нет
Netscape	6.x+	Да
Opera	6.0, 7.0 Macintosh и Windows	Да
Safari	1.2	Да

Несмотря на неадекватную поддержку, многие дизайнеры пропагандируют использование именно этого приема всюду, где возможно, поскольку его можно считать стандартизованным методом горизонтального центрирования в CSS.

Центрирование с помощью свойства text-align

Другой способ центрирования состоит в использовании свойства `text-align`. Ему нужно присвоить значение `center` и применить к элементу body. Это, конечно, отъявленное мошенничество, но оно работает во многих браузерах, так что трюк можно взять на вооружение.

Мошенничеством мы назвали этот прием потому, что свойство текста применяется к объемлющему элементу, а не к тексту. А это создает дополнительные сложности.

Создав `div`, необходимые для макета, вы применяете свойство `text-align` следующим образом:

```
body {
  text-align: center;
}
```

В чем же здесь проблема? А в том, что и все потомки body тоже будут центрироваться (рис. 4).

А значит, придется писать дополнительные правила для корректировки и восстановления стандартного отображения текста:

```
p {
  text-align: left;
}
```

Как вы понимаете, это не слишком удобно. Кроме того, по-настоящему совместимые со стандартом браузеры не будут выравнивать контейнер, только текст.

Сочетание задания полей с выравниванием текста

Принимая во внимание проблемы с обратной совместимостью свойства text-align и желая воспользоваться поддержкой автоматических полей в большинстве современных браузеров, многие дизайнеры объединяют оба подхода, для того чтобы центрирование правильно работало во всех браузерах:

```
body {
  text-align: center;
}
#container {
  margin-left: auto;
  margin-right: auto;
  border: 1px solid red;
  width: 168px;
  text-align: left;
}
```

Увы, такое решение тоже несовершенно и по-прежнему остается трюком. Приходится писать дополнительные правила для управления выравниванием текста. Но все же оно дает хорошие результаты для большинства браузеров.

Решение на основе отрицательных полей

Столкнувшись с проблемой центрирования в «Энтомологии», Хикс решил применить еще один прием: задать отрицательные значения полей.

Впрочем, для центрирования недостаточно просто задать отрицательные поля. Понадобится еще применить абсолютное позиционирование.

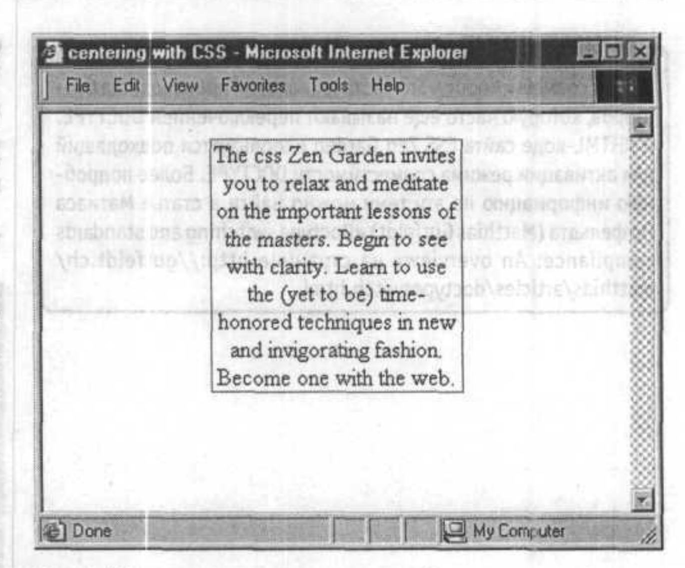


Рис. 4 ✦ Центрирование с помощью свойства text-align в Internet Explorer. Обратите внимание, что текст абзаца внутри div тоже центрирован

ПРИМЕЧАНИЕ

Важно подчеркнуть, что отрицательные значения полей в CSS вполне законны. С их помощью можно решить целый ряд проблем дизайна, например задачу о перекрытии прямоугольников, содержащих некоторые элементы.

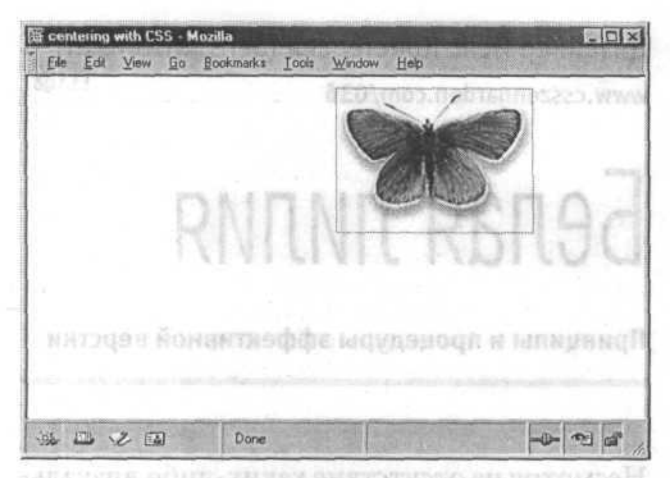


Рис. 5 ✦ Абсолютное позиционирование объемлющего блока со смещением 50% от левого края

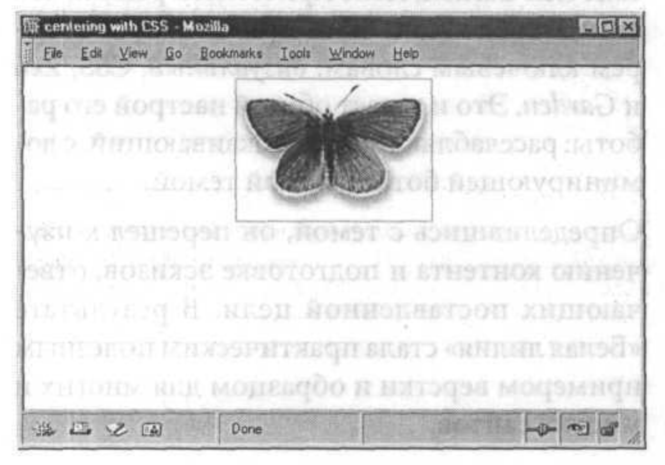


Рис. 6 ✦ Если для элемента задано отрицательное поле, равное половине его общей ширины, то элемент оказывается точно в центре

ПРИМЕЧАНИЕ

Будьте внимательны к арифметике! При вычислении ширины нужно учитывать отступы и границы, иначе нужный эффект не будет достигнут.

Вот как это делается. Сначала создаем контейнер, абсолютно позиционируем его и задаем отступ от левой границы – 50 процентов. В результате левое поле объемлющего блока будет начинаться в середине страницы (рис. 5).

Затем левому полю контейнера присваивается отрицательное значение, равное *половине* ширины объемлющего элемента. В итоге объемлющий блок располагается точно по центру (рис. 6).

Вот фрагменты из таблицы стилей «Энтомологии»:

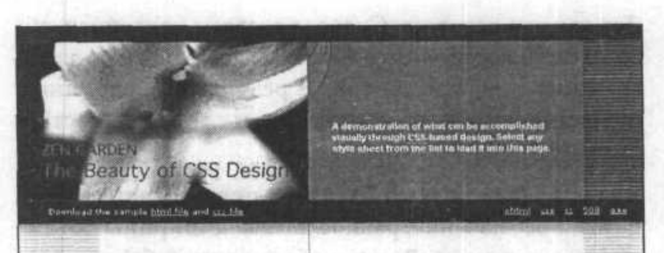
```
#container {
  background: #ffc url(mid.jpg) repeat-y
center;
  position: absolute;
  left: 50%;
  width: 760px;
  margin-left: -380px;
}
```

Ловкость рук и никакого мошенства! Хотя этот метод нельзя назвать предпочтительным, но в качестве обходного пути он годится и поддержан очень широко. Вы не поверите, но даже Netscape 4.x поддерживает эту технику, поэтому она оказывается исключительно полезной, если дизайн должен работать с максимально большим числом разных браузеров.

Верность стилю

Мода приходит и уходит, а дизайн с фиксированной версткой останется еще надолго. Он дает возможность очень точно определить макет страницы и потому находится в арсенале любого дизайнера.

Какой подход предпочесть, зависит от личных пристрастий и того, на какие браузеры вы ориентируетесь. Использование для центрирования комбинации отрицательных полей и абсолютного позиционирования (как в «Энтомологии») – это прием, согласующийся со спецификацией и работающий в самых разных браузерах, в том числе и тех, от которых трудно ожидать поддержки.



Дизайнер Йенс Кристенсен (Jens Kristensen)

www.csszengarden.com/036

Белая лилия

Принципы и процедуры эффективной верстки

Несмотря на отсутствие каких-либо визуальных ограничений для работ, присылаемых на сайт Zen Garden, Йенс Кристенсен решил, что внимание должно быть привлечено к четырем ключевым словам: *визуальный*, CSS, *Zen* и *Garden*. Это и задает общий настрой его работы: расслабляющий, успокаивающий, с доминирующей ботанической темой.

Определившись с темой, он перешел к изучению контента и подготовке эскизов, отвечающих поставленной цели. В результате «Белая лилия» стала практическим полезным примером верстки и образцом для многих и многих сайтов.

ПРИМЕЧАНИЕ

Знакомые с другими видами графических искусств, например, с дизайном печатных изданий или упаковки, очень скоро приходят к выводу, что у макета, предназначенного для отображения на экране, имеется ряд уникальных особенностей. Начинающие заниматься веб-дизайном быстро осознают, что, хотя у верстки для Web есть ряд общих черт с макетированием печатной страницы, тем не менее это совершенно разные вещи.

Принципы верстки для Web

Мы часто, не задумываясь, употребляем слово *верстка* или *макет*, понимая под ним просто число колонок на странице. Люди, изучающие CSS, часто именно так и подходят к верстке, задавая вопрос: «Как создать макет с тремя колонками?» или «Как создать макет с заголовком и хвостовиком?». Это правильные и важные вопросы, но еще до того, как приступать к CSS, надо ясно представлять себе принципы, следуя которым мы выбираем тот или иной макет.

Принципы эффективной верстки веб-страниц корнями уходят во времена, когда никакой Всемирной паутины не было и в помине. И хотя для Web эти принципы претерпели изменения, но они остаются основой для создания макетов, которые не только приятны глазу, но и хорошо функционируют.

Во всех видах графического искусства макетирование служит одной главной цели: эффективно организовать контент. Зрительное представление макета зависит от носителя, природы контента и аудитории. Вторичная задача, встающая именно перед веб-дизайнером, проистекает из того факта, что макет должен еще и поддерживать определенные функции. Речь идет о ссылках, системах меню и других функциональных элементах, которые должны быть интегрированы в дизайн. И наконец, макет – это краеугольный камень единства сайта, без которого им трудно будет пользоваться.

Семь раз отмерь, один раз отрежь

Создавая «Белую лилию», Кристенсен для начала попытался организовать контент, с которым ему предстояло работать.

Организация контента преследует несколько целей. У вас складывается четкое представление о природе материала, а следовательно, и о сравнительной важности тех или иных фрагментов. Далее вы начинаете размышлять о макете и различных подходах, которые помогли бы добиться нужной формы и функциональности.

Первым делом Кристенсен выделил три категории контента на сайте CSS Zen Garden:

- ✦ общая информация, к примеру, ссылки на разные шаблоны, инструменты проверки и обзорную информацию о сайте;
- ✦ ссылки для навигации по самому сайту Zen Garden;
- ✦ основной контент.

Располагая этой информацией, он приступил к работе над версткой страницы. Теперь у него было две вполне конкретных цели: реализовать свою тему и представить контент, принадлежащий трем разным категориям. Для выработки окончательного решения потребовалось несколько итераций (рис. 1).

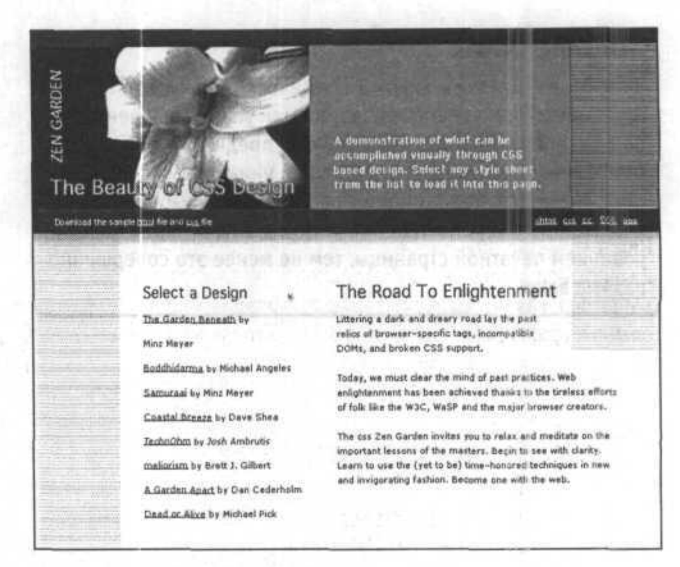


Рис. 1 ✦ Альтернативный дизайн «Белой лилии»

В конце концов внешний облик «Белой лилии» прояснился. Важно, чтобы при выборе макета вы основывались на природе контента и визуальных требованиях, а не принимали решение наугад или следуя текущей моде.

Решения, принятые при создании макета «Белой лилии»

Имея четкое представление о том, что надо делать, Кристенсен должен был решить, как подойти к визуальному представлению темы и контента.

Автор решил, что фотография лилии задаст нужный ему настрой на ботаническую тему. Фотография также определила цветовую гамму. Кроме того, он разработал дополнительный декор, подчеркивающий растительные мотивы (рис. 2).

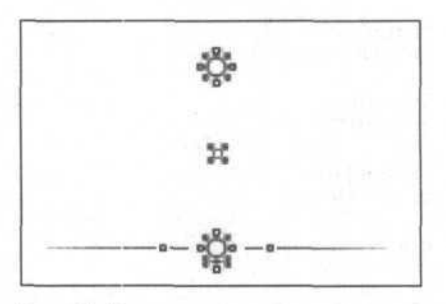


Рис. 2 ✦ Три стилизованных цветка, подчеркивающих ботаническую тему «Белой лилии»

Но по-настоящему работа Кристенсена заиграла, когда дело дошло до контента. Поскольку цель была четко поставлена, то не возникло сложностей с определением относительной важности контента и выбором эффективного представления.

Поместив общую информацию о саде в заголовок, Кристенсен достиг двух важных целей: представить «логотип» (в данном случае цветы и название страницы CSS Zen Garden) и рассказать о причинах появления данной страницы. Кроме того, в этот же раздел помещены ссылки на примеры HTML и CSS-файлов, а равно на инструменты проверки (рис. 3).

Download the sample html file and css file — xhtml css cc 508 aaa

Рис. 3 ✦ Решение одной из важнейших задач: эффективное группирование взаимосвязанного контента



Рис. 4 ✦ Метафора – это важный аспект проектирования интерфейса. Изображение магазинной тележки, такой, например, как на сайте amazon.com, – это общепринятый символ, адаптированный к нуждам электронной коммерции

Select a Design

The Final Ending
by Ray Henry

contemporary nouveau
by David Hellsing

El Collar de Tomas
by Maria Stultz

OrderedZen
by Steve Smith

Bonsai
by Martin Plazotta

Type Thing
by Michal Mokrzycki

Pseudo Sahara
by John Barrick

Dragen
by Matthew Buchanan

Archives

next designs »
View All Designs

Resources

View This Design's CSS
CSS Resources
FAQ
Submit a Design
Translations

Рис. 5 ✦ Навигационные ссылки в дизайне «Белая лилия»

Оставшееся место освободилось для навигации и основного контента. Элементы навигации Кристенсен поместил в левую колонку, поскольку именно там посетитель ожидает их найти. Правая же колонка используется для контента. Оба раздела умеренно стилизованы и не загромождены излишними украшениями. Сочетание заголовка и двухколонной верстки прекрасно удовлетворяет поставленным требованиям, получившийся несложный дизайн *работает*.

Рекомендации по верстке

Сформулировав общие цели, вы можете воспользоваться нижеследующими рекомендациями для детальной верстки. Мы приведем некоторые общие принципы, применимые для любых макетов веб-страниц. В основе их лежит удобство веб-интерфейса и практичность.

Проектирование интерфейса пользователя

В отличие от печатных изданий, веб-страница предполагает интерактивное взаимодействие, а не просто пассивное созерцание. Дизайн интерфейса пользователя – это относительно неплохо разработанная область, поскольку интерактивные интерфейсы применяются в самых разных компьютерных технологиях, от операционных систем до прикладных программ.

Вот некоторые фундаментальные принципы проектирования пользовательского интерфейса:

- решите, на какую категорию пользователей будет ориентирован интерфейс;
- стремитесь к тому, чтобы пользователь не заблудился на странице и на сайте в целом;
- используйте знакомые метафоры. Прекрасным примером может служить символ магазинной тележки на сайте электронной коммерции. Даже незнакомые с Web люди сразу понимают его смысл (рис. 4);
- старайтесь, чтобы фунциональные особенности были ясно видны. В «Белой лилии» это списки ссылок, назначение которых очевидно с первого взгляда (рис. 5);
- будьте последовательны в выборе элементов дизайна. Если бы «Белая лилия» была взята в качестве шаблона для полноценного сайта, то элементы навигации следовало бы расположить одинаково на всех страницах, равно как фотографию и логотип. Изменение местоположения эмблемы сайта и функциональных элементов может запутать пользователя, избегайте таких решений;

- следите за центром внимания. В дизайне всегда должен быть какой-то объект, притягивающий взгляд. В «Белой лилии» это сам цветок. Выявив центральный объект, дизайнер может затем расположить остальные элементы для достижения зрительного баланса;
- не оставляйте сомнений в том, чему же посвящена страница. Люди не могут долго сохранять сосредоточение; обычно они согласны ждать не более 9 секунд – если за это время человек не поймет, в чем дело, то, скорее всего, переключится на что-то другое. Кристенсен добивается внимания посетителя за счет размещения основного контента в отдельной области на странице (рис. 6.).

Дизайнеры интерфейсов всегда озабочены соотношением эстетики и функциональности. Обратите внимание на стрелочки в заголовках разделов списка ссылок в «Белой лилии». Все они одинакового размера, имеют приятный внешний вид и помогают посетителю сориентироваться (рис. 7).

Практичность

Хотя концептуально практичность и дизайн пользовательского интерфейса связаны между собой (и их исторические корни одинаковы), но цели их несколько различаются. Пользовательский интерфейс естественным образом касается пользователей, поэтому главная задача инженеров и разработчиков – понять, как лучше удовлетворить их потребности.

При изучении практичности упор делается на том, каким образом пользователь работает с предложенным интерфейсом. Оба вопроса тесно связаны друг с другом, поскольку результаты исследования практичности влияют на методы проектирования интерфейсов, а разработанные интерфейсы предлагаются пользователям для тестирования с целью последующего улучшения.

Как и проектирование пользовательских интерфейсов, изучение практичности – кстати, тоже пришедшее из других компьютерных технологий – ведется уже давно. Практичность в Web – эта лишь часть более широкой проблематики, но для нее выработаны специфические рекомендации.

Вот некоторые предложения по вопросам верстки, имеющим отношение к практичности:

- Информация должна быть «на холме». Конечно, в применении к Web «холм» – это лишь метафора, но смысл ее ясен: та информация, которая нужна пользователю в первую очередь, должна располагаться как можно ближе к началу страницы. В «Белой лилии» логотип сайта и заголовок страницы сразу бросаются в глаза (рис. 8).

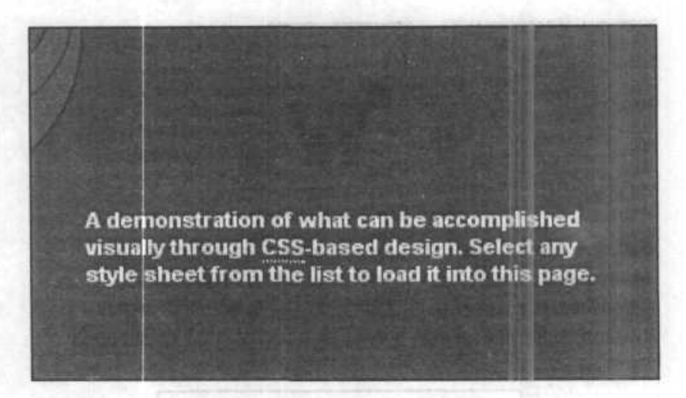


Рис. 6 ✦ Если вы поместите главный элемент в отдельную область, посетитель сможет быстро понять назначение страницы

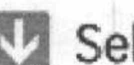


Рис. 7 ✦ Стрелки в заголовках разделов списка ссылок эстетичны и вместе с тем функциональны, поскольку привлекают внимание посетителя к расположенному ниже контенту

> **ПРИМЕЧАНИЕ**
> Проектированию пользовательского интерфейса и интерактивному взаимодействию посвящен замечательный сайт Boxes and Arrows (www.boxesandarrows.com).



Рис. 8 ✦ Заголовок, содержащий важную информацию о странице, находится в самом верху

css Zen Garden: The Beauty in CSS Design

Рис. 9 ✦ Описание в заголовке окна хотя и не является частью верстки, но помогает посетителю ориентироваться на сайте

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see submission guidelines), but we ask you release your CSS under a Creative Commons license identical to the one on this site so that others may learn from your work.

Рис. 10 ✦ Информация об авторском праве или лицензионной политике – это важный элемент контента, который обязательно должен быть предусмотрен в верстке

ПРИМЕЧАНИЕ

Во многих макетах информация об авторских правах, обеспечении конфиденциальности, лицензии и т.п. размещается в хвостовике. Этого общепринятого соглашения следует придерживаться при проектировании сайтов, для которых такого рода информация – существенная сторона деятельности организации или компании.

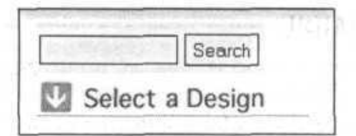


Рис. 11 ✦ Если бы в этом дизайне присутствовало поле для ввода поискового запроса, то находиться оно могло бы над ссылками в области навигации

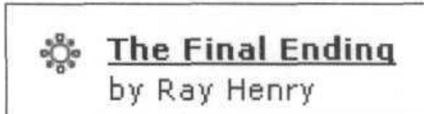


Рис. 12 ✦ Интересно, что отступы при размещении текста в колонке помогают лучше понять функциональность. В данном случае отступы применяются в главном списке ссылок, визуально разбивая колонку на разделы, из-за чего весь макет становится более свободным

- Всегда размещайте описание в элементе `title`:

  ```
  <title>css Zen Garden: The Beauty of the CSS
  Design</title>
  ```

  Эта строка помещает указанную строку в заголовок окна браузера, что помогает пользователю понять, о чем идет речь на странице (рис. 9).
- Следите за тем, чтобы ссылки и система навигации размещались по возможности в одних и тех же местах. Кроме того, включайте ссылку на начальную страницу. Полезны также ссылки на страницы с описанием авторских прав, политики обеспечения конфиденциальности и контактной информации. В случае сайта CSS Zen Garden все варианты дизайна защищены лицензией Creative Commons, что четко обозначено в тексте (рис. 10).
- Для большинства средних и больших сайтов настоятельно рекомендуется реализовать систему поиска. Понятно, что CSS Zen Garden – это демонстрация подходов к дизайну, а не методов построения сайтов, но на практике пользователи ценят возможность поиска и часто ею пользуются. Если, конечно, она расположена в интуитивно понятном месте. Как правило, поле для ввода поискового запроса находится там, где ему логически надлежит быть, например, рядом с элементами навигации. Если бы в «Белой лилии» была предусмотрена возможность поиска, то поместить ее следовало бы, наверное, над списком ссылок (рис. 11).
- Разбивайте содержимое колонок с помощью отступов. Это способствует улучшению внешнего облика страницы и помогает пользователю уяснить ее структуру (рис. 12).

Конечно, о практичности можно говорить еще долго, но следование даже этим рекомендациям позволит выделить важные детали, касающиеся формы и функции, и в конечном счете сделать работу со сверстанной вами страницей более удобной для пользователя.

Следуйте процедуре

Дизайн «Белая лилия» демонстрирует, как следование четко определенной процедуре приводит к решению многих вопросов верстки. Конечно, различные дизайнеры и коллективы строят процесс работы по-разному, и возможно, ваш подход существенно отличается.

Но какой бы ни была конкретная процедура, ясно, что, определив сначала тему и затем проанализировав контент, Кристенсен нашел очень короткий путь к цели: создать дизайн, приятный и удобный для пользователя.

Дизайнер **Минц Мейер** (Minz Meyer)

www.csszengarden.com/037

Прет-а-порте

Горизонтальные линии в вертикальном мире

Начинающие дизайнеры чаще всего задают вопрос: «Откуда черпать идеи?» Им советуют разное. К примеру, в природе есть немало необычных ландшафтов, способных стать источником вдохновения и свежих идей.

Другой источник вдохновения – человек. Форма человеческого тела всегда занимала особое место в искусстве: студенты тратят немало времени в студиях, изучая рельеф тела, черты лица и другие особенности человека. В дизайне «Прет-а-порте» Минца Мейера природа и человек создают уникальное сочетание, впечатление от которого усиливается необычной версткой.



Печатная страница как источник вдоховения

Веб-дизайнеры, уже проработавшие в этой области некоторое время, часто слышат от маститых авторов и преподавателей одно и то же: «Web – не то же самое, что печатное издание». Конечно, это так, но дизайн «Прет-а-порте» убеждает нас в том, что, хоть веб-страница и печатная страница – вещи разные, но нет причин, по которым дизайн последней не мог бы стать источником вдохновения для веб-дизайнера.

В данном случае идеей «Прет-а-порте» стала не просто печатная страница, а страница, *посвященная печати* (рис. 1).



Рис. 1 ✦ Фрагмент печатной брошюры, из которой родилась идея «Прет-а-порте»

В оригинальной брошюре Мейера, созданной как реклама типографии, для создания визуального напряжения применялось сочетание графических образов и слов. Левая часть была отведена под вертикальный ряд фотографий – стилизованных для данной брошюры и снятых в один день – женского лица необычной расцветки. Справа помещены изображения со словами и фразами, в частности, с немецким словом, переводящимся как «обесцвеченный». В результате брошюра подчеркивает качество работы типографии с помощью ярких образов и ясной терминологии, подводя читателя к выводу, что эта типография не допустит ошибок при выборе цветов для печатных изображений.

Оригинальная брошюра была сверстана в формате «ландшафт», то есть по горизонтали, а не по вертикали. Именно в этом заключается самый интригующий и интересный аспект макета соответствующей веб-страницы.

В нарушение всех правил

Хотя ландшафтный формат эффектно выглядит в печатных изданиях и потому применяется очень часто, но веб-дизайнерам до последнего времени рекомендовали избегать такой верстки. Причины связаны главным образом с практичностью. В случае горизонтальной верстки браузер помещает полосу прокрутки вдоль нижнего края документа. Это считалось неудобным, поскольку пользователи обычно не любят

совершать движений больше, чем это необходимо для получения информации. А для людей с некоторыми физическими недостатками прокручивать окно по горизонтали особенно трудно. Поэтому веб-дизайнеры, как правило, прибегали к горизонтальной верстке лишь в особых случаях, скажем, в целях эксперимента или, что особенно печально, создавали такого рода дизайн по ошибке, неправильно оценив разрешающую способность экрана.

Редким исключением стал запущенный в 1998 году сайт Circle of Friends (В кругу друзей), посвященный людям, страдающим от болезней позвоночника. Эта организация, основанная в числе прочих ныне покойным Кристофером Ривом (Christopher Reeve), выбрала такой дизайн, потому что именно из горизонтального положения люди с больным позвоночником вынуждены смотреть на мир. В то время подобная дизайнерская концепция была революционной, но отражала конкретный человеческий опыт (рис. 2).

Заинтересовавшись тем, как горизонтальное расположение материала может повлиять на Web, Мейер решил попробовать создать убедительную версию печатной страницы – не только работоспособную, но и сверстанную с применением CSS.

Горизонталь бросает вызов

Попытка объединить глубокое эмоциональное воздействие оригинальной брошюры с версткой на основе CSS натолкнулась на проблему поддержки со стороны браузеров. Чтобы дизайн выглядел эстетично и был удобен для работы в разных браузерах, приходилось искать различные обходные пути.

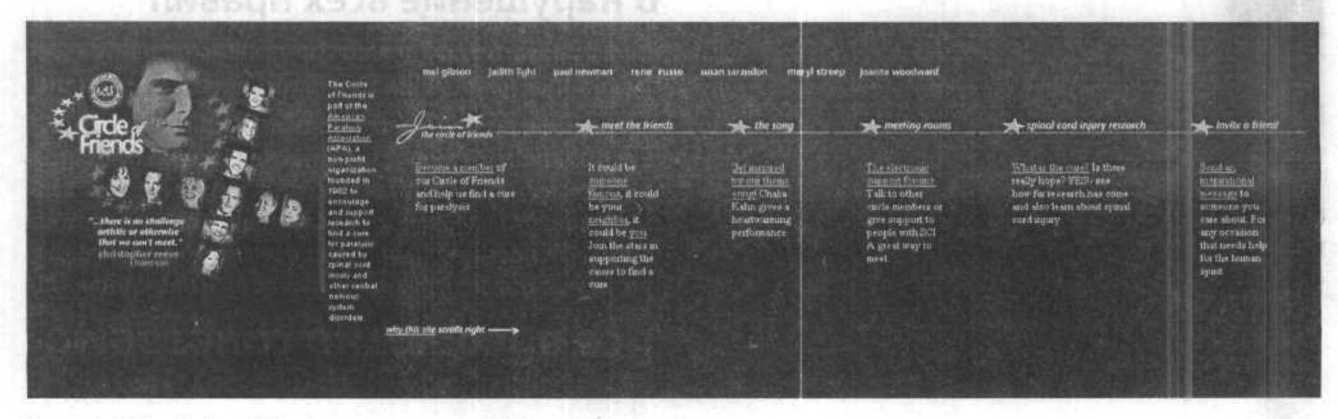


Рис. 2 ✦ Сайта Circle of Friends, запущенного в 1998 году, уже нет, но это один из ранних примеров нарушения запрета на горизонтальную верстку

Фиксированное позиционирование

Яркий образ женщины побудил Мейера разместить картинку так, чтобы она всегда была видна на экране, а текст прокручивался под ней. Этот эффект возможен только в Web.

CSS предоставляет дизайнеру много возможностей для реализации фоновой графики, но если сделать фотографию женщины фоновой картинкой в элементе body, то текст будет прокручиваться поверх нее. При этом и текст будет невозможно читать, и воздействие образа сведется на нет.

Один из способов решить эту проблему – присоединить изображение к div с фиксированной позицией. Элемент с фиксированной позицией будет всегда оставаться в одном и том же месте на странице, он не прокручивается, что бы ни происходило с остальными элементами. Для этой цели подошел бы такой селектор:

```
div#static-image {
  background-image: url(bg_face.jpg);
  background-repeat: no-repeat;
  background-position: left bottom;
  position: fixed;
  left: 0;
  bottom: 0;
  height: 594px;
  width: 205px;
  z-index: 2;
}
```

Если браузер полностью совместим со спецификацией CSS, то значение fixed свойства position гарантирует, что этот div будет жестко закреплен на странице, а прочие элементы можно будет расположить за ним (рис. 3).

СОВЕТ

Свойство z-index позволяет задать порядок перекрытия элементов. Элемент с наибольшим значением z-index окажется поверх всех остальных.

Рис. 3 ✦ В браузере Mozilla, который поддерживает фиксированное позиционирование, изображение остается на одном месте, перекрывая прокручиваемый контент



Ошибки в браузерах

Как это часто бывает, когда приходится учитывать дефекты браузеров, разумно сначала создать идеал, а затем заняться браузерами, которые по той или иной причине не поддерживают нужного эффекта. Для этого создаются разные варианты дизайна, а затем с помощью JavaScript-сценария определяется, с каким браузером мы имеем дело. Впрочем, некоторые считают такой подход неэффективным, в частности потому, что многие пользователи отключают поддержку JavaScript в своих браузерах.

В случае «Прет-а-порте» курс был взят на то, чтобы создать альтернативный дизайн для Internet Explorer, который бы хорошо выглядел и прорисовывался, пусть даже изображение лица прокручивалось бы за пределы окна.

Internet Explorer не поддерживает также селекторов дочерних элементов. Синтаксически такие селекторы задаются с помощью символа >. Так, селектор `p>strong` означает, что следующее за ним объявление должно быть применено только к элементам `strong` внутри абзацев.

В связи с отсутствием поддержки фиксированного позиционирования и селекторов дочерних элементов в Internet Explorer, были написаны правила, которые реализуют оптимальный дизайн для браузеров, совместимых со спецификацией CSS, и альтернативный дизайн для Internet Explorer.

```
body#css-zen-garden>div#extraDiv2 {
  background-image: url(bg_face.jpg);
  background-repeat: no-repeat;
  background-position: left bottom;
  position: fixed;
  left: 0;
  bottom: 0;
  height: 594px;
  width: 205px;
  z-index: 2;
}
```

Следующее правило применяется только в браузерах, не понимающих предыдущего, в данном случае в Internet Explorer. Поскольку специфичность селекторов дочерних элементов больше, то у предыдущего правила более высокий приоритет, но лишь в том случае, если браузер его понимает.

ПРИМЕЧАНИЕ

Хотя значение `fixed` для свойства `position` было введено в спецификации CSS2 еще в 1998 году, но некоторые браузеры, прежде всего Microsoft Internet Explorer, его до сих пор не поддерживают. Поэтому, хотя созданная таблица стилей прекрасно работает в более совместимых браузерах, в Internet Explorer страница прокручивается целиком, что сводит на нет весь задуманный эффект.



Рис. 4 ✦ В альтернативном варианте дизайна для Internet Explorer 6.0 картинка прокручивается за пределы окна, но целостность дизайна все же сохраняется



Рис. 5 ✦ Так выглядит «Прет-а-порте» в браузере Opera 7.0. В версиях Opera младше 7.5 не появлялась полоса горизонтальной прокрутки, из-за чего большая часть страницы была недоступна

```
div#extraDiv2 {
  background-image: url(bg_face_ie.jpg);
  background-repeat: no-repeat;
  background-position: left bottom;
  position: absolute;
  left: 0;
  bottom: 0;
  height: 600px;
  width: 265px;
}
```

И хотя картинка с фиксированным позиционированием в Internet Explorer 6.0 уходит за пределы окна, дизайн все-таки остается эстетичным и приемлемым (рис. 4).

Особенность браузера Opera

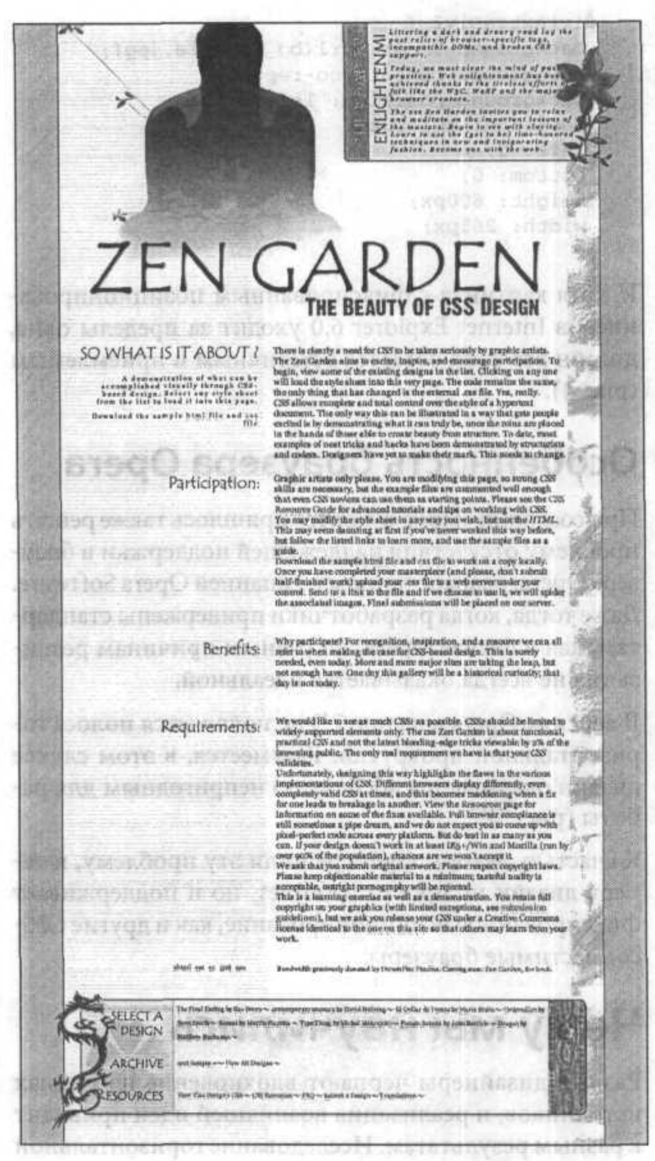
При создании «Прет-а-порте» пришлось также решать проблему отсутствия надлежащей поддержки в браузере Opera, разработанном компанией Opera Software. Даже тогда, когда разработчики привержены стандартам, как в случае Opera, по разным причинам реализация не всегда оказывается идеальной.

В версиях Opera младше 7.5 не появляется полоса горизонтальной прокрутки. Разумеется, в этом случае дизайн оказывается абсолютно непригодным для работы (рис. 5).

К счастью, авторы Opera решили эту проблему, и теперь дизайн не просто работает, но и поддерживает фиксированное позиционирование, как и другие CSS-совместимые браузеры.

Чему мы научились

Разные дизайнеры черпают вдохновение из разных источников, и реализация возникшей идеи приводит к разным результатам. Исследование горизонтальной верстки, предпринятое Мейером в связи с разработанным им дизайном печатной брошюры, выявило ряд новых возможностей и проблем. Применив тонкие обходные пути, Мейер сумел создать удивительно красивый дизайн, который не только подсказывает, где искать плодотворные идеи, но и демонстрирует их эффективную реализацию.



Дизайнер **Седрик Саварезе** (Cedric Savarese)

www.csszengarden.com/070

Монах

Выразительное позиционирование и сетки

Кто сказал, что сначала нужно составить план? Дизайнер Седрик Саварезе работал над своим «Монахом», экспериментируя в Adobe Photoshop и наблюдая, что получается. Достоинством такого подхода является гибкость: когда в голову приходит новая мысль, старые можно отбросить или приспособить к новому варианту верстки.

Саварезе решил, что внимание нужно привлечь в первую очередь к заголовку, а остальной макет выстроить вокруг него. Перепробовав множество шрифтов, он выбрал для заголовка и подзаголовка те, которые в наибольшей степени соответствовали идее дизайна. Тема дзен-буддизма была дополнена садовыми мотивами, а для визуальной привлекательности использована спокойная красно-серая цветовая гамма.

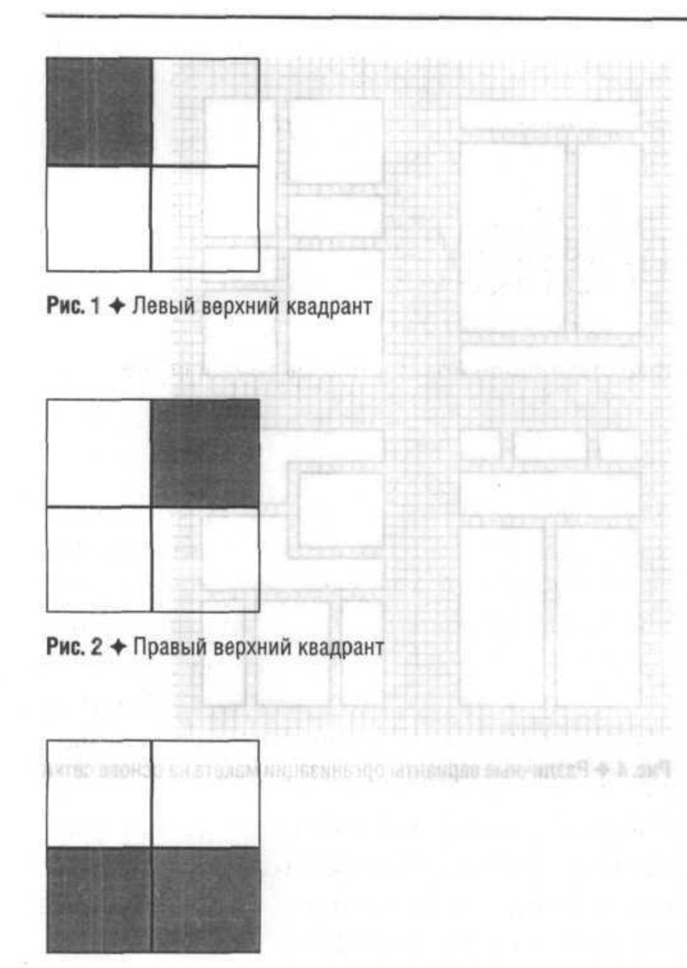


Рис. 1 ✦ Левый верхний квадрант

Рис. 2 ✦ Правый верхний квадрант

Рис. 3 ✦ Левый нижний и правый нижний квадранты

Позиционирование как выразительное средство

Доминирующее изображение монаха занимает место, на которое зритель обращает внимание в первую очередь, – левый верхний угол. Самый первый элемент дизайна – одновременно самый идейно важный и занимает наиболее значимое место.

Если взять плоский «холст» фиксированного размера, например лист бумаги, и расчертить его пополам по вертикали и по горизонтали, то получится четыре квадранта. В западных языках текст читается слева направо и сверху вниз. Поэтому типичный зритель бессознательно приписывает разным квадрантам разные приоритеты.

Вы никогда не задумывались, почему логотипы сайтов обычно находятся в левом верхнем углу? Да потому, что именно туда прежде всего устремляется взор посетителя. Поместив какой-либо элемент в левый верхний угол, мы привлечем к нему максимум внимания, поэтому для торговых марок – это самое подходящее место (рис. 1).

Правый верхний квадрант тоже важен (рис. 2). На веб-странице выделяется пространство, расположенное «на холме», то есть видимое без прокрутки. Именно ему придается наибольшее значение. Элемент, находящийся в правом верхнем углу, чуть менее важен, чем тот, что занимает левый верхний угол, но все же значим для страницы в целом.

Два нижних квадранта не слишком существенны для большинства веб-страниц (рис. 3). Некоторые дизайнеры помещают сюда дополнительную навигацию, другие – лишь информацию об авторских правах и больше ничего. В традиционном дизайне печатных изданий размещение элемента внизу страницы может создать впечатление, будто расположенные выше элементы приподняты. Такое ощущение трехмерности характерно только для печатных страниц, а в Web на информацию, находящуюся внизу, просто не обращают внимания.

Все шире распространяется интересная практика размещения малозначимой информации вдоль нижнего края страницы. Здесь могут оказаться ссылки на внутренний контент или на функции сайта, не нашедшие места среди механизмов регулярной навигации. Если посетитель не заметит их, ничего страшного, он сможет добраться до нужного контента другими способами. Ну а те, кто обратят внимание на такие ссылки, получат возможность воспользоваться дополнительными функциями.

Сетка

Сетка – это важный инструмент верстки, но из-за «негибкости» ею часто пренебрегают. Как и в математике, сеточный макет – это просто набор прямоугольников, на которые окно разбивается перпендикулярными прямыми. Именно в эти прямоугольники и помещаются позже элементы дизайна. Если строго следовать сетке, то пропорции и промежутки получатся более выверенными, правда, ценой чрезмерной упорядоченности и сухости (рис. 4).

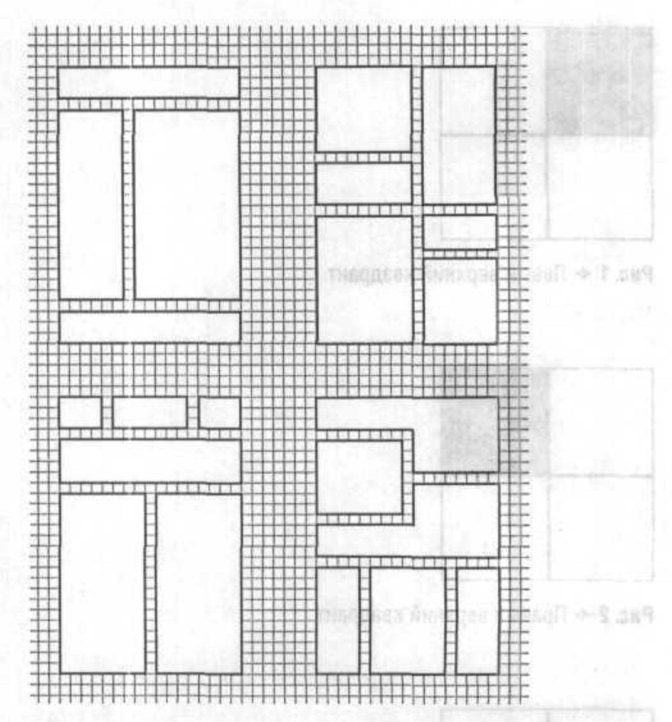


Рис. 4 ✦ Различные варианты организации макета на основе сетки

Классический табличный дизайн ярко демонстрирует ограничения, налагаемые излишне структурированной сеткой, – у элемента, помещенного в ячейку таблицы, нет другого выхода, как только оставаться в этой ячейке. Если необходим эффект перекрытия, то нужно задействовать соседние клетки. Часто единственный способ как-то нарушить регулярность состоит в разбиении изображения на мелкие кусочки, которые затем собираются путем создания сложных вложенных таблиц.

С другой стороны, CSS позволяет гораздо точнее контролировать расположение элементов, не прибегая к помощи сеток. В основе макета может лежать сеточная структура, но размещение отдельных элементов с нарушением регулярности осуществляется гораздо более гибко. Модель позиционирования, принятая в CSS, обеспечивает элементам куда большую свободу. Элементы могут располагаться за пределами своих контейнеров, да и вообще в любом месте страницы.

Построение сетки

Конечно, чтобы выйти за пределы сетки, ее надо сначала построить. Если наложить сетку на «Монаха», то окажется, что заголовок разбит на три компонента, а основная область занимает две колонки.



Рис. 5 ✦ Силуэт монаха и заголовок объединены в одну картинку

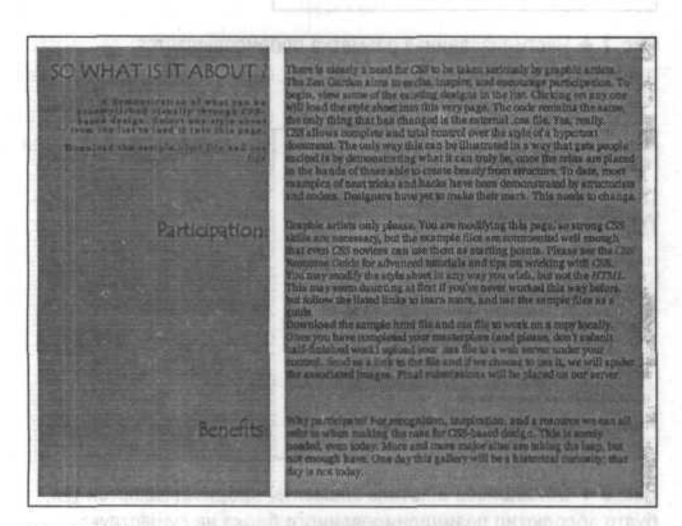


Рис. 6 ✦ Кажущаяся вторая колонка выровнена по сетке

Заголовок

Заголовок состоит из трех областей, важность которых неодинакова: силуэта монаха, текста Zen Garden и текста из раздела `#preamble`, который вынесен в заголовок из нормального потока документа.

Но при взгляде на отдельные картинки, составляющие заголовок «Монаха», создается впечатление, что силуэт и текст объединены в одно изображение. Хотя на самом деле это разные структурные части заголовка, при кодировании они рассматриваются как единое целое. Это вообще типично для веб-дизайна – перед сохранением изображения надо объединить в одну картинку, чтобы потом было проще кодировать (рис. 5).

Область контента

Хотя в основе дизайна «Монах» лежит одна центральная колонка с текстом, но заголовки, помещенные на полях, создают ощущение, будто есть вторая колонка. Мы раскрасили дизайн в синий цвет, чтобы были наглядно видны пропорции обеих колонок (рис. 6).

Ширина левой колонки составляет примерно 2/3 ширины правой, то есть соблюдается соотношение 2:3. Пропорции устанавливаются автоматически: коль скоро ширина первой колонки задана, вторая занимает все оставшееся место. А отношение 2:3 – это классическое гармоничное отношение, которое определяет естественный и приятный ритм страницы.

Подробнее о пропорциях и отношениях см. книгу Kimberley Elam «Geometry of Design: Studies in Proportion and Composition» (Princeton Architectural Press, 2001).

Выход за пределы сетки

Саварезе выстроил основную часть макета «Монаха» на базе сетки, но некоторые элементы расположил вне ее ячеек, чтобы нарушить регулярность и сделать макет более интересным.

Гибкости имеющейся в CSS модели позиционирования для этого достаточно, причем можно воспользоваться как абсолютным, так и относительным позиционированием. Анализ различий между ними позволит принять решение, какой вариант лучше применить.

Абсолютное позиционирование

Чтобы разобраться с абсолютным позиционированием, надо понимать, что такое поток документа. Мы уже писали, что язык HTML по природе своей линеен: сначала может идти элемент `h1`, за ним `p`, потом `div`. Элементы могут быть вложенными, но если не применяются никакие стили, то HTML-документ прорисовывается последовательно с начала и до конца (рис. 7).

Абсолютное позиционирование позволяет не только переместить элемент в любое место на странице, но и изъять его из нормального потока документа. Абсолютно позиционированный блок не влияет ни на какие другие элементы, как будто его никогда и не было в потоке (рис. 8). Визуально он окажется в том месте, куда его поместили, но с точки зрения других элементов его просто не существует. Решать вопросы, связанные с возможным перекрытием, – задача дизайнера.

Единственный случай, когда абсолютно позиционированный блок как-то влияет на другой элемент, – это если последний является потомком первого и также позиционирован. Согласно модели позиционирования, любой элемент позиционируется относительно своего контейнера. В большинстве случаев контейнером служит корневой элемент документа html. Однако если у рассматриваемого элемента есть позиционированный предок, отличный от корневого элемента, то позиционирование производится относительно него.

Подробнее о контейнерах см. спецификацию CSS на странице «10 Visual formatting model details» (www.w3.org/TR/REC-CSS2/visudet.html#containing-block-details).

css Zen Garden

The Beauty of *CSS* Design

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the *W3C*, *WaSP* and the major browser creators.

Рис. 7 ✦ Нестилизованная разметка прорисовывается линейно сверху вниз

css Zen Garden

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

The Beauty of *CSS* Design

Download the sample html file and css file

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the *W3C*, *WaSP* and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to

Рис. 8 ✦ Оставшиеся в потоке элементы прорисовываются так, будто абсолютно позиционированного блока не существует

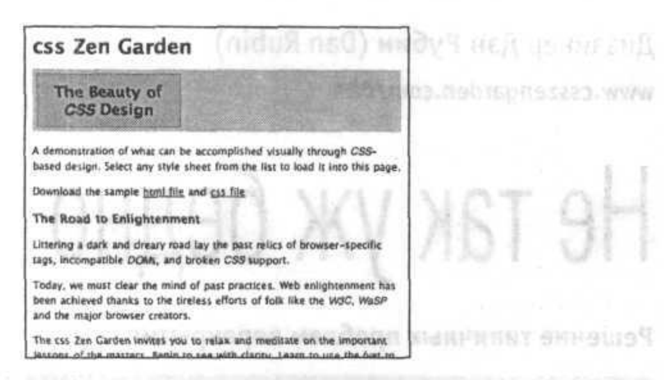


Рис. 9 ✦ Элемент в нормальном потоке документа

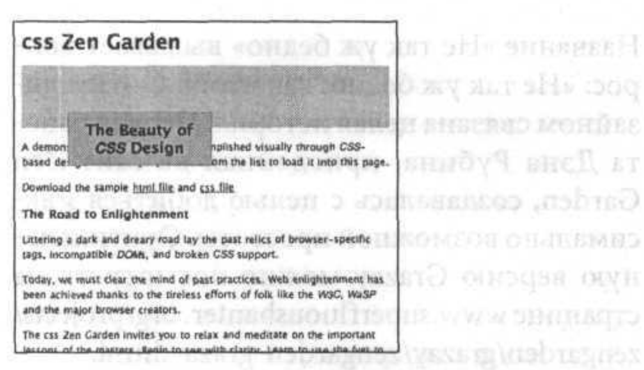


Рис. 10 ✦ Относительно позиционированный элемент остается в потоке; при вычислении положения других элементов учитывается его начальная позиция, а окончательная игнорируется

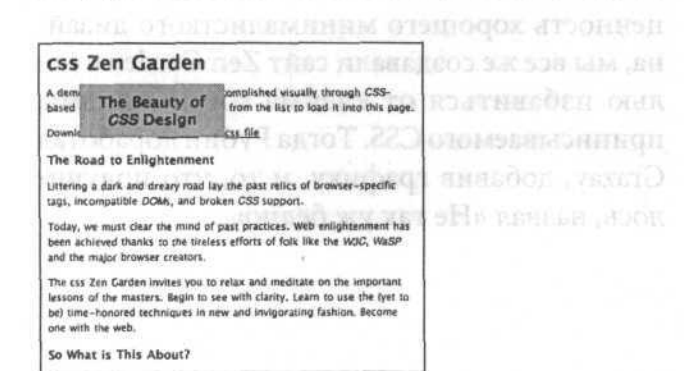


Рис. 11 ✦ Напротив, абсолютно позиционированный элемент не оказывает никакого влияния на нормальный поток документа. Игнорируется не только его новая, но даже и начальная позиция

Относительное позиционирование

В случае относительного позиционирования элемент не изымается из нормального потока. Сначала определяется естественная позиция элемента, а затем элемент сдвигается относительно нее. Таким образом, на странице остается «дырка», которую другие элементы должны, тем не менее, принимать во внимание (рис. 9, 10 и 11).

Относительное позиционирование чаще всего применяется для сдвига элементов; элемент, первоначально расположенный в ячейке сетки, можно легко вынести за ее пределы. Можно также осуществить тонкую подгонку, когда никакие другие способы позиционирования неприменимы. Например, Саварезе с помощью относительного позиционирования переместил дракона из его естественной позиции под белой областью контента в левый угол хвостовика. Применить абсолютное позиционирование к элементам, находящимся в нижней части макета, гораздо труднее, чем к элементам, расположенным ближе к началу, поэтому в данном случае использование относительного позиционирования позволяет решить задачу более простым способом.

Гибкость и регулярность

Процедура «эволюционного дизайна», примененная Седриком Саварезе, может привести к хаосу, если не знать меры. Понимание модели позиционирования и пропорций позволит избежать лишних проблем, а наложение на макет сетки – добиться гармоничного результата.



Select any style sheet from the list to load it into this page. Download the sample HTML FILE and CSS FILE

→ SELECT A DESIGN
- The Final Ending
- contemporary nouveau
- El Collar de Tomas
- OrderedZen
- Bonsai
- Type Thing
- Pseudo Sahara
- Dragen

→ ARCHIVES
- next designs »
- view all designs

→ RESOURCES
- view this design's css
- css resources
- faq
- submit a design
- translations

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved *thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.*

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for *CSS* to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

PARTICIPATION

Graphic artists only please. You are modifying this page, so strong *CSS* skills are necessary, but the example files are commented well enough that even *CSS* novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with *CSS*.

You may modify the style sheet in any way you wish, but not the *HTML*. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. Send us a link to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

BENEFITS

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

REQUIREMENTS

We would like to see as much *CSS1* as possible. *CSS2* should be limited to widely-supported elements only. The css Zen Garden is about functional, practical *CSS* and not the latest bleeding-edge tricks viewable by *2% of the browsing public. The only real* requirement we have is that your *CSS* validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid *CSS* at times, and this becomes maddening when a fix for one leads to breakage in another. View the Resources page for information on some of the fixes available. Full browser compliance is *still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect* code across every platform. But do test in as many as you can. *If your design doesn't* work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see submission guidelines), but we ask you release your *CSS* under a Creative Commons license identical to the one on this site so that others may learn from your work.

Bandwidth graciously donated by DreamFire Studios. Coming soon: *Zen Garden, the book.*

XHTML CSS CC 508 AAA

Дизайнер **Дэн Рубин** (Dan Rubin)
www.csszengarden.com/024

Не так уж бедно

Решение типичных проблем перекрытия

Название «Не так уж бедно» вызывает вопрос: «Не так уж бедно, как что?». С этим дизайном связана целая история. Первая работа Дэна Рубина, присланная на сайт Zen Garden, создавалась с целью добиться максимально возможной простоты. Оригинальную версию Grazay можно посмотреть на странице www.superfluousbanter. org/projects/zengarden/grazay/zengarden-grazay.html.

Рубин прислал вариант Grazay (рис. 1), выбрав очень простую цветовую гамму и шрифты (FF DIN Condensed и Verdana). Признавая ценность хорошего минималисткого дизайна, мы все же создавали сайт Zen Garden с целью избавиться от клейма минимализма, приписываемого CSS. Тогда Рубин доработал Grazay, добавив графику, и то, что получилось, назвал «Не так уж бедно».



Рис. 1 ✦ Оригинальный дизайн Grazay показывает, по сравнению с чем вариант «Не так уж бедно» оказался не таким уж бедным

ПРИМЕЧАНИЕ

Сочетание дизайна, основанного на стандартах, с контентом, генерируемым системами CMS, наталкивается и еще на одну проблему: многие популярные программы порождают некорректную разметку, включающую, помимо прочего, устаревшие презентационные теги. Применение стилей к тегу `font` – это бессмыслица, а заказать разработчику специализированные доработки обычно не по карману большинству потребителей.

Когда контента слишком много

При верстке страницы с помощью CSS очень часто приходиться бороться с неопределенностью, присущей Web. Точнее, проблема состоит в том, как быть с предопределенными значениями ширины и высоты в CSS в случае, когда объем контента заранее неизвестен.

Неопределенность длины контента – неотъемлемая особенность веб-страниц, которые могут содержать как много, так и мало информации. Программы, управляющие динамическими сайтами (например, системы управления контентом – CMS), могут вставлять контент переменной длины – совсем короткий или весьма и весьма пространный.

Если ни автор, ни система CMS не налагают жестких ограничений на длину контента – а обычно это нереально, то нужно обязательно проверять, как будет выглядеть любая содержащая контент область при разной его длине. Поведение текстового контента менее предсказуемо, чем графического, так как размеры изображений известны заранее и не изменяются. Но даже это не является непреложным правилом, поскольку CMS может вставлять несколько изображений или изображения разной высоты. Да и вообще есть множество факторов, приводящих к отличиям в размерах.

Впрочем, ситуация постепенно улучшается, последние версии CMS генерируют корректную разметку в формате XHTML/CSS. К числу наиболее распространенных систем управления контентом можно отнести Drupal (www.drupal.org) и Mambo (www.mamboserver.com).

Многие современные браузеры предоставляют пользователю возможность изменять масштаб текста, и люди со слабым зрением этим активно пользуются. При увеличении размера шрифта тоже возможно нежелательное перекрытие. Если занятая контентом область устроена так, что проблемы с переменной длиной текста не возникает, то теоретически не должно быть и сложностей из-за масштабирования. Но короткие фрагменты текста, например навигационные ссылки, часто страдают от этой проблемы. Мы рекомендуем всегда проверять, выглядит ли страница приемлемо при увеличении размера шрифта на 150% и даже больше, поскольку обязательно найдутся пользователи, которые захотят так сделать.

Рано или поздно вы, скорее всего, столкнетесь с ситуацией, когда область контента переполняется по вертикали или по горизонтали. В результате вы увидите либо перекрывающийся контент, либо область, простирающуюся далеко вниз, – гораздо дальше, чем следовало бы. Первое указывает на некорректное абсолютное позиционирование, второе – свидетельство наличия плавающих объектов, слишком высоких или широких для отведенной под них области.

Переполнение в плавающих объектах

Строго говоря, атрибут `float` – это не инструмент верстки. Плавающие объекты предназначены для того, чтобы текст мог обтекать строчные элементы, их применение для верстки макета никогда не рассматривалось. Но нам приходится пользоваться плавающими элементами по одной простой причине: нет другого способа расчистить место (clear) для контента.

В простейшей форме атрибут `clear` и вся концепция расчистки сводятся к тому, чтобы сказать браузеру, что некий элемент не следует начинать рисовать, пока не будет завершен какой-то другой элемент. Проблема переполнения при абсолютном позиционировании по большей части проистекает из невозможности расчистить место, занятое позиционированным объектом.

Но механизм плавающих объектов не идеален, и его несовершенства еще ярче проявляются из-за неодинаковой поддержки браузерами.

Для расчистки нужна подходящая разметка

Чтобы механизм расчистки правильно работал, в HTML-коде должны присутствовать по крайней мере два элемента: тот, к которому применяется свойство `float`, и еще один, который расчищает место. Два `div` могут «уплыть» соответственно налево и направо, но тогда понадобится третий, который прекратит обтекание. Часто можно воспользоваться уже существующими в разметке элементами, например, прекратить обтекание плавающего `h3` с помощью одного из следующих за ним абзацев `p`.

Но иногда приходится мириться с неизбежным злом: добавлять лишний пустой элемент в разметку с единственной целью расчистить место. Для таких случаев мы рекомендуем универсальный `div`, не содержащий никакого контента:

```
<div class="clear"></div>
```

СОВЕТ

Эрик Мейер написал большую статью на тему плавающих объектов «Containing floats», обязательно прочитайте ее (www.complexspiral.com/publications/ containing-floats).

СОВЕТ

Существует также метод расчистки места за плавающими объектами, не требующий вставки специального элемента в HTML-разметку. Правда, чтобы он работал во всех браузерах, придется воспользоваться несколькими интуитивно неочевидными трюками. См. статью «How to Clear Floats Without Structural Markup» на странице www.positioniseverything.net/easyclearing.html.



Рис. 2 ✦ Если плавающие элементы оказываются слишком широкими для отведенной под них области, то один или несколько из них могут быть вытеснены и окажутся ниже остальных

К этому классу может быть применено свойство `clear`. Решение некрасивое, но оно работает и не слишком загромождает структурную разметку ненужными элементами.

При размещении плавающих объектов нужна точность

Если наполнить аквариум до краев, а потом всыпать в него горсть песку, что произойдет? Понятно что – часть воды выльется.

У плавающих объектов аналогичные ограничения, связанные по большей части с переполнением по горизонтали. Если два плавающих объекта, каждый шириной в 50% окна, расположить один за другим и задать для первого поле шириной 100 пикселей, то второй будет вытеснен и окажется под первым. Поэтому нужно заранее рассчитывать ширину.

Но Microsoft Internet Explorer для Windows не поддерживает атрибуты `width` и `height` так, как это рекомендовано в спецификации CSS. Предполагается, что элемент, контент которого шире, чем значение `width`, попытается перерисовать свой контент, а если это не получится, то оставит ширину неизменной и позволит контенту «выплеснуться» за границы контейнера. Смысл в том, чтобы не менять положение плавающего объекта, даже если контент начинает перекрывать другие элементы.

Но Internet Explorer позволяет элементу расширяться по мере необходимости. Как только плавающий объект становится хотя бы на пиксель шире допустимого, он вытесняет другие плавающие объекты (рис. 2). Решить эту проблему можно несколькими способами.

Сделать область шире

Если есть возможность увеличить ширину области, в которой размещены плавающие объекты, то этого может оказаться достаточно. Такое решение не оптимально, но начать нужно с него, поскольку реализуется оно очень просто.

Ограничить ширину контента

Если можно гарантировать, что контент никогда не вызовет переполнения по горизонтали, то проблемы удастся избежать. Обычно такое решение нереалистично, поскольку если бы контент не был слишком широким, то проблемы бы вообще не возникло.

Иногда, чаще всего в Internet Explorer для Windows, встречается одна особенно глупая ошибка, из-за которой контент, казалось бы, имеющий нормальную ширину, все-таки вызывает переполнение. Если вы с этим столкнетесь, посмотрите, нет ли в области, занятой плавающим объектом, текста, отображаемого курсивом. Если есть, попробуйте убрать курсив и посмотрите, не решит ли это проблему. В Internet Explorer курсив может приводить к переполнению, и если его отключение помогает, то виновник найден. Вообще-то никакое форматирование текста не должно вызывать такого эффекта, но жизнь есть жизнь.

Обработать переполнение

В следующем разделе мы рассмотрим свойство `overflow` в применении к переполнению по вертикали, но с его помощью можно обработать и переполнение по горизонтали тоже. Впрочем, подобное решение зачастую выглядит неэстетично, так что мы рекомендуем не злоупотреблять им.

Переполнение при абсолютном позиционировании

При абсолютном позиционировании чаще всего возникают проблемы с переполнением по вертикали. Исключение составляют случаи, когда для позиционированного элемента задана ширина, а отведенная под него область оказывается недостаточно широкой (рис. 3).

Рис. 3 ✦ Элемент `#linklist` шириной 150 пикселей не помещается в отведенную под него область шириной 100 пикселей

Переполнение по вертикали

Сталкиваться с переполнением по вертикали приходится куда чаще. Проблема проста: за абсолютно позиционированными элементами нельзя расчистить место, как за плавающими. Если элемент позиционирован, то он изымается из нормального потока документа и перестает оказывать влияние на своих соседей.

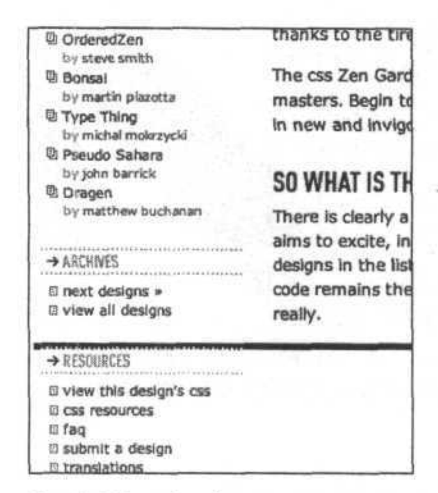


Рис. 4 ✦ Если бы область, занятая контентом, была короче, то абсолютно позиционированная левая колонка в дизайне «Не так уж бедно» наехала бы на хвостовик

Такая неприятность часто возникает при размещении хвостовиков страниц – абсолютно позиционированная колонка может оказаться длиннее других колонок, определяющих положение хвостовика. В этом случае позиционированная колонка «наедет» на хвостовик (рис. 4). К сожалению, настоящего решения этой проблемы не существует, только обходные пути.

Пользуйтесь плавающими объектами

Самый эффективный обходной путь – пользоваться плавающими объектами вместо абсолютно позиционированных и существующим для них механизмом расчистки места. Конечно, это не всегда возможно, но коль скоро верстка позволяет применить плавающие объекты, рассмотрите этот вариант.

Попробуйте изменить дизайн

Вместо этого можно дать абсолютно позиционированному элементу расти вниз по вертикали без ограничений. Если он перекрывает хвостовик, то можно перепроектировать последний, ограничив его ширину. Если ниже позиционированного элемента ничего нет, то и перекрывать нечего.

Обработка переполнения

Чем пытаться управлять длиной контента, гораздо удобнее воспользоваться механизмом автоматической обработки переполнения, встроенным в CSS. Свойство `overflow` было специально придумано для этой цели, но назвать его «серебряной пулей» все же нельзя. Если ограничить высоту элемента 150 пикселями и указать для свойства `overflow` значение `auto`, то область будет обрезана точно по указанной высоте, а для просмотра не поместившегося контента пользователь получит полосу прокрутки.

Однако слишком большое число полос прокрутки на одной странице раздражает пользователей. Можно вместо `auto` задать значение `hidden`, тогда непоместившийся контент вообще не будет показан. Но, как

правило, скрывать контент таким образом не стоит, поскольку невидимый контент бесполезен для читателя. Впрочем, бывают ситуации, когда такой метод – как раз то, что нужно.

Воспользуйтесь сценарием

Если вы допускаете минимальное применение JavaScript в дополнение к таблицам стилей, то решить проблему можно с помощью сценария. Один из авторов Zen Garden, Шон Инман (Shaun Inman), предложил полезный метод расчистки места за абсолютно позиционированными элементами, аналогичный свойству `clear` для плавающих объектов. Его сценарий приведен на странице Absolute Clearance (www.shauninman.com/mentary/past/ absolute_clearance.php).

Задавайте размеры в em, а не в px

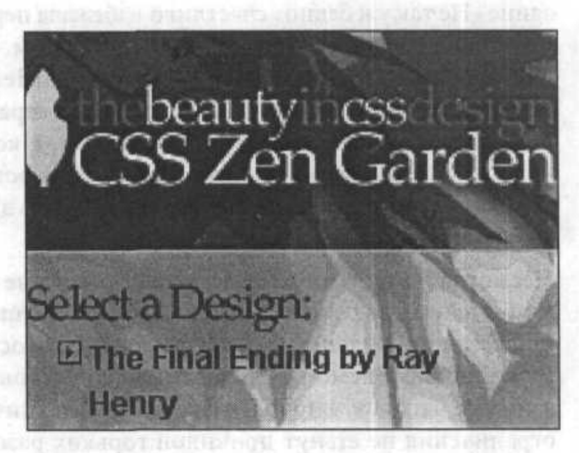
Этот совет применим только в случае, когда перекрытие возникает лишь из-за увеличения размера шрифта; перекрытия вследствие динамического роста длины контента он не излечит. Абсолютное позиционирование с точностью до пикселя обеспечивает максимальный контроль, но страдает от отсутствия гибкости – раз уж элемент позиционирован, то останется точно там, куда его поместили. Задание же для свойств `top` и `left` значений, выраженных в единицах em, дает чуть больше свободы, поскольку позиция будет зависеть от размера шрифта. Если размеры элементов, расположенных один под другим, заданы в em, то при сдвиге нижнего края верхнего элемента сдвигается и верхний край нижнего. Этому методу недостает точности, но во многих ситуациях он работает.

Дизайн, учитывающий возможность переполнения

Абсолютно позиционированная левая колонка в дизайне «Не так уж бедно» счастливо избежала переполнения, поскольку область, занятая контентом, оказалась гораздо длиннее. Если бы Zen Garden был большим сайтом, то могли бы встретиться страницы, на которых левая колонка была длиннее, а контент короче. Тогда бы все неприятности, проиллюстрированные на рис. 4, вылезли наружу, и пришлось думать, как решить проблему переполнения.

Несмотря на то что технические сложности не должны бы влиять на решения, принимаемые дизайнером, при создании больших сайтов переполнение остается серьезной проблемой. Если вы сумеете построить дизайн так, чтобы эту проблему обойти, то технические ограничения не станут причиной горьких разочарований.

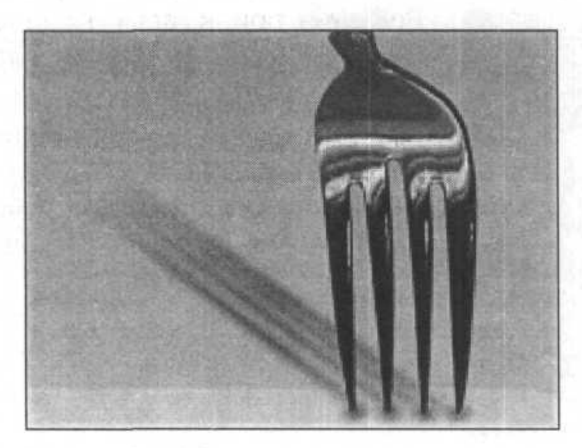
5 Зак. 1942



Японский сад, с. 132



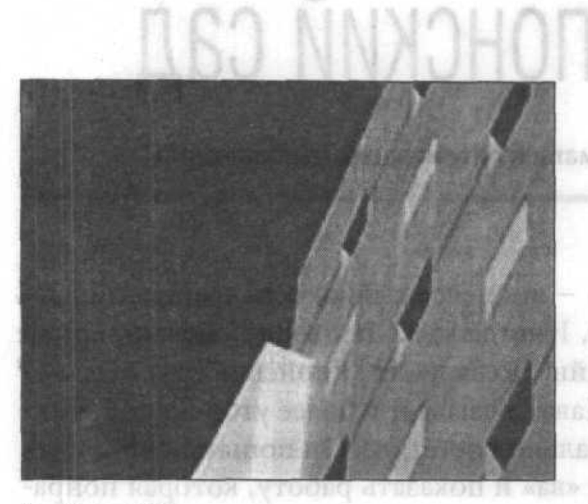
Революция!, с. 140



Без границ!, с. 152



Береговой бриз, с. 158



Арт Деко, с. 146



Что там в глубине, с. 166

4

Графика

Создание графических образов, оказывающих максимально сильное воздействие на зрителя, – важнейшая часть работы дизайнера. Рассмотренные в этом разделе варианты дизайна были отобраны за их выдающуюся графику. Решения, принимаемые в процессе работы над изображением, могут приводить к самым разным результатам.

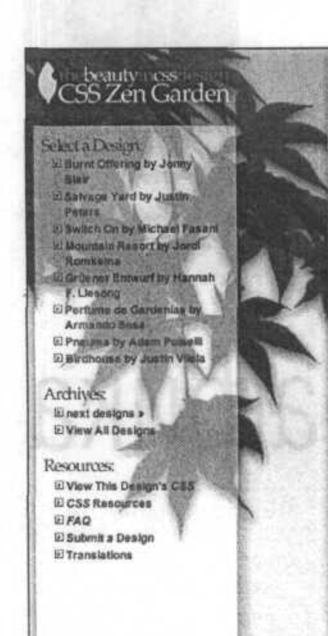
Вдохновляйтесь, изучайте методы манипулирования изображениями, пополняйте свои знания о библиотеках фотографий, учитесь создавать собственные визуальные элементы, когда того требует ситуация.



Дизайнер **Масанори Кавачи** (Masanori Kawachi)
www.csszengarden.com/096

Японский сад

Форматы и оптимизация изображений



A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

悟りへの道 The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the *W3C*, *WaSP* and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

何のためにあるの？ So What is This About?

There is clearly a need for *CSS* to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, *the only thing that has changed is the external* .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

参加 Participation

Graphic artists only please. You are modifying this page, so strong *CSS* skills are necessary, but the example files are commented well enough that even *CSS* novices can use them as starting points. Please see the *CSS* Resource Guide for advanced tutorials and tips on working with *CSS*.

You may modify the style sheet in any way you wish, but not the *HTML*. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. *Once you have* completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. Send us a link to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

特典 Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for *CSS*-based design. This is sorely needed, even today. More and more *major* sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

必要条件 Requirements

We would like to see as much *CSS1* as possible. *CSS2* should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we *have is that your CSS* validates.

Unfortunately, designing this way highlights the flaws in the various implementations of *CSS*. Different browsers display differently, even completely valid *CSS* at times, and this becomes maddening when a fix for one leads to breakage in another. View the Resources page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see submission guidelines), but we ask you release your CSS under a Creative Commons license identical to the one on this site so that others may learn from your work.

Bandwidth graciously donated by DreamFire Studios.

xhtml css cc 508 aaa

«Ва» – *японское слово, означающее гармонию.* Иногда здесь подходит смелый, яркий дизайн. Но вариант «Японский сад» Масанори Кавачи взывает к более утонченной, универсальной эстетике. Он попытался уловить суть «ва» и показать работу, которая понравилась бы представителям самых разных национальностей.

Бросающиеся в глаза японские мотивы в выбранной Кавачи теме указывают на его происхождение, но мягкость визуальных эффектов и обаятельность графики произведут впечатление на любого зрителя, откуда бы он ни был родом.

Графические форматы

Достичь изысканной прозрачности «Японского сада» можно, только хорошо разбираясь в имеющихся графических форматах. Из тысяч разных форматов, применяемых в компьютерной графике, лишь три имеют непосредственное отношение к Web: GIF, JPEG и PNG. Все три позволяют сжимать изображение, чтобы уменьшить его размер и тем самым ускорить загрузку.

Почему так важна оптимизация

Зачем нужно оптимизировать изображение? Ведь широкополосные каналы связи доступны повсеместно, к чему уж так-то скредничать? Конечно, высокоскоростной доступ в Интернет получает все более широкое распространение, но пока еще не стал общедоступным. Многие пользователи по-прежнему работают с модемами, дающими скорость не более 56 Кбит/с, а в некоторых странах за пользование телефонными линиями взимают плату. Каждый лишний килобайт увеличивает время загрузки, а каждая минута стоит реальных денег.

ПРИМЕЧАНИЕ

В США к середине 2004 года свыше 50% домашних пользователей были подключены к Интернету по широкополосному каналу.

Не будем забывать и о сервере. На сайты с высоким трафиком заходит много пользователей; чем большую долю полосы пропускания потребляет каждый посетитель, тем шире должна быть общая полоса пропускания, доступная веб-серверу, а значит, увеличивается плата за хостинг. Каждый килобайт, сэкономленный на графическом файле, оборачивается экономией денег как пользователей, так и владельцев сайта. 2 Кб тут, 2 Кб там – вроде немного, но в итоге выливается в солидную сумму. К примеру, если урезать на 2 Кб картинку, которую каждый день просматривают 100000 посетителей, то общая экономия составит 6 Гб в месяц.

Какой формат выбрать

В разных графических форматах применяются разные цветовые модели, разные виды сжатия и разные формы достижения прозрачности. Чтобы выбрать подходящий формат, нужно знать сильные и слабые стороны каждого.

В случае *сжатия без потери информации* сохраняются все детали изображения, при *сжатии же с потерей информации* часть деталей утрачивается ради уменьшения размера файла. У каждого метода есть свои области применения.

Глубиной цвета характеризуют максимальное число цветов в изображении. Глубина цвета измеряется в *битах*, чем больше бит используется для представления цвета, тем больше общее число доступных цветов. Если глубина цвета равна 8 бит, то всего может быть 256 цветов, при глубине цвета 24 бит общее число цветов превышает 64 миллиона.

Формат GIF

Дедушка всех графических форматов – GIF – существует в той или иной форме уже почти 20 лет. В нем применяется метод сжатия без потери информации, называемый *LZW*, лучше всего он подходит для сохранения изображений, в которых есть обширные области, закрашенные одним цветом.

Хотя в формате GIF информация сохраняется без потерь, но в нем используется так называемый индекс цвета, а это палка о двух концах. В одном изображении может быть не больше 256 цветов, поэтому применение этого метода ограничено лишь изображениями, содержащими мало цветов. Зато для изображений, содержащих меньше 256 цветов, можно использовать меньший индекс цвета, что уменьшает окончательный размер файла (рис. 1–3). Изображения с большим числом цветов *можно* сохранить в формате GIF, но ценой утраты информации о цвете, в результате чего мы приходим к сжатию с потерей информации.

Формат GIF поддерживает 1-битовую прозрачность. Это означает, что пиксель может быть либо полностью прозрачным, либо полностью непрозрачным. Если вы привыкли к графическому редактору, работающему со слоями и обеспечивающему многоуровневую альфа-прозрачность, то подобное ограничение раздражает. Но даже такой режим полезен, если заранее известно, на каком фоне будет отображаться GIF-файл. Кроме того, формат GIF поддерживает простую покадровую анимацию и постепенную загрузку (эта техника называется *черезстрочной разверткой*). Правда, и то и другое приводит к увеличению размера файла.



Рис. 1 ✦ Изображение сохранено в формате GIF с 256 цветами, размер файла равен 8,6 Кб



Рис. 2 ✦ Изображение сохранено в формате GIF с 32 цветами, размер файла равен 5,0 Кб



Рис. 3 ✦ Изображение сохранено в формате GIF с 4 цветами, размер файла равен 2,1 Кб

ПРИМЕЧАНИЕ

Цветной вариант рис. 1–3 см. на вкладке.



Рис. 4 ✦ Изображение сохранено в формате JPEG с коэффициентом качества 70, размер файла равен 8,3 Кб



Рис. 5 ✦ Изображение сохранено в формате JPEG с коэффициентом качества 30, размер файла равен 3,7 Кб



Рис. 6 ✦ Изображение сохранено в формате JPEG с коэффициентом качества 10, размер файла равен 2,5 Кб



Рис. 7 ✦ Изображение сохранено в формате PNG с глубиной цвета 24 бит, размер файла равен 16,2 Кб



Рис. 8 ✦ Изображение сохранено в формате PNG с глубиной цвета 8 бит и 64 цветами, размер файла равен 5,3 Кб



Рис. 9 ✦ Изображение сохранено в формате PNG с глубиной цвета 8 бит и 4 цветами, размер файла равен 2,2 Кб

ПРИМЕЧАНИЕ

Цветной вариант рис. 4–9 см. на вкладке.

Формат JPEG

Благодаря 24-битовой глубине цвета формат JPEG отлично подходит для сохранения фотографий и других многоцветных детальных изображений. JPEG допускает сжатие с потерей информации, если вы готовы пожертвовать толикой деталей ради уменьшения размера файла (рис. 4–6).

При сильном сжатии в изображении будут видны отчетливые диспропорции и мелкие дефекты отдельных пикселей, называемые *артефактами*. Хотя в более детальном изображения эти дефекты не так заметны, все же при использовании формата JPEG надо соблюдать баланс между приемлемым ухудшением качества картинки и уменьшением размера файла. В JPEG прозрачность вообще не реализована, но JPEG с прогрессивной разверткой, функционально аналогичный GIF с чересстрочной разверткой, поддерживается большинством браузеров (см. ниже раздел «Приемы оптимизации»).

Формат PNG

В этом формате, во многих отношениях похожем на GIF, есть два режима хранения цветов. PNG-файлы с 24-битовой глубиной цвета велики по размеру, пожалуй, даже слишком велики, зато не приводят к потере информации. Сохраняются абсолютно все детали. В файлах же с 8-битовой глубиной, как и в GIF, применяется индекс цвета, и во многих случаях степень сжатия оказывается даже выше, чем при сохранении того же изображения в формате GIF с эквивалентными параметрами (рис. 7–9).

Благодаря наличию двух режимов глубины цвета формат PNG теоретически можно использовать для хранения любых изображений; на практике же лучше ограничиться только графикой с небольшим числом цветов, для которой обычно применяется формат GIF, а изображения фотографического качества сохранять в формате JPEG. Хотя в формате PNG с 24-битовой глубиной цвета информация не теряется, но достигается это ценой увеличения размера файла.

Поддерживается также два режима прозрачности: 1-битовый и 8-битовый. Первый не отличается от того, что есть в GIF, второй же обеспечивает все 256 уровней альфа-канала. 8-битовый режим прозрачности интересен тем, что можно заранее не думать, на каком фоне будет отображаться PNG-файл. К сожалению, полезность этого режима ограничена в силу слабой поддержки

со стороны браузеров. В частности, Microsoft Internet Explorer для Windows плохо справляется с альфа-прозрачностью в PNG-файлах.

Вопросы прозрачности

Несмотря на наличие огромного числа графических форматов, в Web нашли применение только три вышеперечисленных. JPEG не поддерживает прозрачность вовсе, в GIF реализована лишь 1-битовая прозрачность, а 8-битовая прозрачность, предлагаемая форматом PNG, не поддерживается большинством современных браузеров. Проблема не нова и не связана непосредственно с CSS. В прошлом десятилетии, чтобы использовать эффект прозрачности в Web, вообще приходилось прорисовывать изображения заранее и сохранять их в том или ином формате, не поддерживающем прозрачность.

В «Японском саде» Кавачи выводит текст в колонках белого цвета, кажущихся прозрачными, поскольку они перекрывают изображение зеленых листьев (рис. 10). Формат GIF не допускает такого эффекта, тем не менее применяется именно он. Вместо того чтобы требовать поддержки прозрачности от браузера, Кавачи воспользовался слоями Adobe Photoshop, и сохранил уже готовые изображения в GIF-файлах. Картинки, выступающие фоном, никак не задействуют прозрачности, именно в таком виде они и были подготовлены. Эффект перекрытия одного цвета другим достигнут лишь умелыми манипуляциями графикой в Photoshop.

При использовании прозрачных GIF-файлов нужно быть особенно внимательными; так как GIF обеспечивает только один уровень прозрачности, то полупрозрачные пиксели должны отображаться на конкретном фоне, именуемом *подложкой* (matte color). Подложка должна быть точно того же цвета, что и фон, на котором будет выводиться GIF-файл на веб-странице, в противном случае будет отчетливо видна каемка из пикселей не того цвета (рис. 11–13).

В Web для создания эффекта прозрачности обычно применяется один из двух рассмотренных выше методов, поскольку они надежны и испытаны временем. Альфа-прозрачность, имеющаяся в PNG, могла бы



Рис. 10 ✦ В «Японском саде» сочетаются непрозрачные и псевдопрозрачные изображения



Рис. 11 ✦ Прозрачный GIF, который был подготовлен для белого фона, отображается на белом фоне



Рис. 12 ✦ Прозрачный GIF, который был подготовлен для белого фона, отображается на светло-зеленом фоне



Рис. 13 ✦ Прозрачный GIF, который был подготовлен для белого фона, отображается на темно-зеленом фоне

ПРИМЕЧАНИЕ

Цветной вариант рис. 11–13 см. на вкладке.

открыть новые возможности, но отсутствие поддержки со стороны Internet Explorer делает эти мечтания тщетными. Или все-таки нет?

Если просто включить прозрачный PNG-файл в дизайн, то в Internet Explorer он будет выглядеть некрасиво, но есть разные хитрости, позволяющие эту проблему обойти. Правда, они работают только для Internet Explorer, но ведь работают же.

Приемы оптимизации

Чтобы получить файл небольшого размера, надо пожертвовать качеством. Но чем сильнее сжимается изображение, тем уродливее оно становится, поэтому лучше начать с высокого уровня сжатия и постепенно уменьшать его, пока качество не окажется приемлемым.

Формат GIF

Для оптимизации GIF вы начинаете с небольшого числа цветов и увеличиваете его до достижения пристойного результата. Начните с 8 цветов, потом попробуйте 16, 32, 64 и так далее, пока не будете удовлетворены. Для некоторых изображений требуется 256 цветов (или даже больше, тогда придется обратиться за помощью к JPEG), другие же не отличить от оригинала даже при использовании всего 8 или 16 цветов.

На размер GIF-файла влияют и другие факторы. Например, с помощью псевдосмешения цветов (dithering) можно заполнить область одного цвета узором из точек близких цветов – в результате достигается иллюзия наличия большего числа цветов, чем есть в действительности. Но применение этой методики снижает эффективность сжатия. Если вместо псевдосмешения просто увеличить число цветов в палитре, то размер файла может получиться даже меньше.

В Photoshop сравнительно недавно появилась возможность сохранять изображения в GIF-файлах с потерей информации. Как бы сильно вы ни сжали файл предварительно, включение этого режима обычно позволяет еще больше сократить размер – конечно, за счет

качества. При очень небольшом коэффициенте потери качества – скажем, 5–6 процентов – размер файла часто удается уменьшить на 20–30 процентов.

Формат JPEG

Оптимизация JPEG – гораздо более простая процедура. В графическом редакторе обычно имеется ползунок, с помощью которого задается уровень качества. Чем выше качество, тем больше файл; чем качество ниже, тем файл меньше. Для начала задайте невысокое качество и постепенно увеличивайте его, пока изображение не будет выглядеть прилично.

Современные браузеры поддерживают и другие возможности формата JPEG. *Оптимизированное* изображение дает чуть более высокий коэффициент сжатия, чем по умолчанию. JPEG с прогрессивной разверткой лучше применять для больших фотографий, при этом размер файла несколько увеличивается. *Размывание* (blur) изображения также позволяет еще уменьшить файл.

ПРИМЕЧАНИЕ

Способы включить прозрачный PNG-файл в состав страницы, которая будет просматриваться в Internet Explorer, описаны в статьях «PNG Behavior» (http://webfx.eae.net/dhtml/pngbehavior/pngbehavior.html) и «Cross-Browser Variable Opacity with PNG: A Real Solution» (www.alistapart.com/articles/pngopacity).

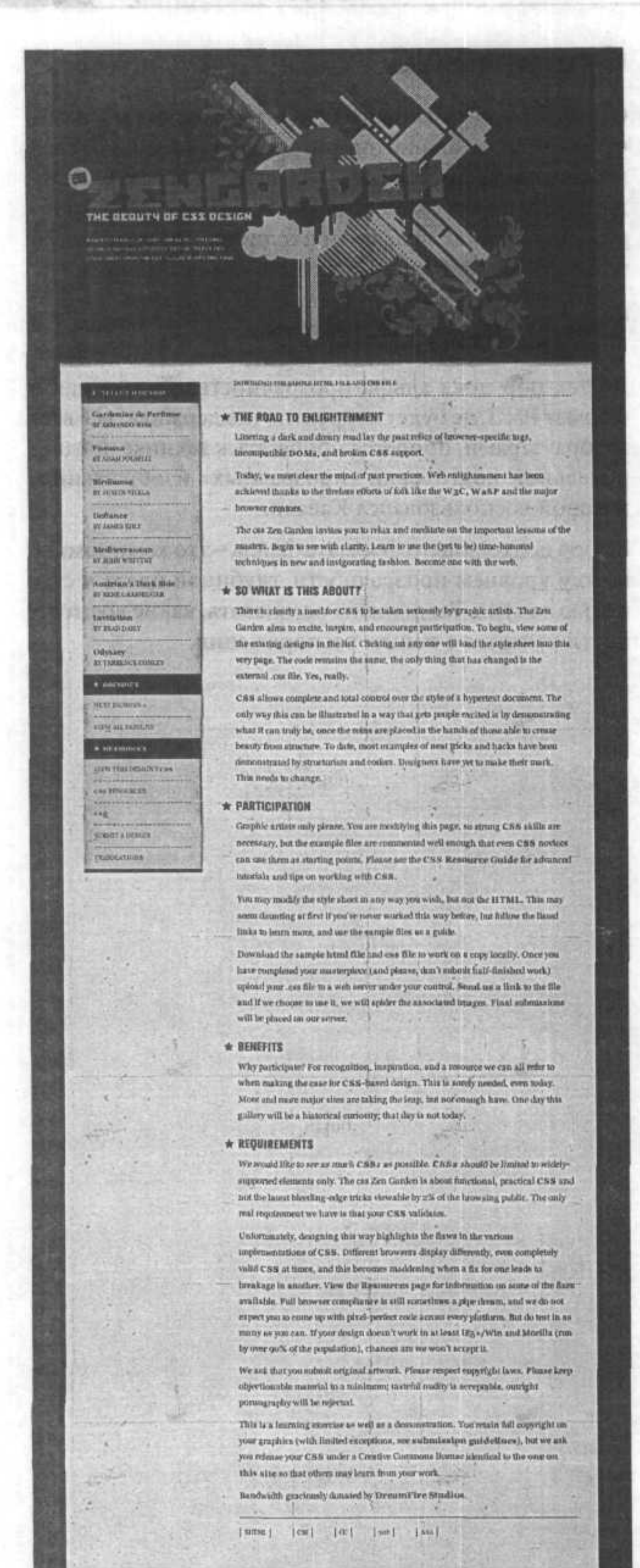
Формат PNG

Оптимизация 8-битового PNG-изображения мало чем отличается от аналогичной процедуры для GIF: вы можете выбрать число цветов и включить режим псевдосмешения. Оптимизировать 24-битовое PNG-изображение вообще нельзя, приходится ограничиваться тем сжатием, которое заложено в самом формате.

Трудный выбор

До тех пор пока альфа-прозрачность, встроенная в формат PNG, не будет корректно поддерживаться всеми браузерами, придется прибегать к технике предварительной подготовки «прозрачных» изображений, которой воспользовался Кавачи.

Выбор подходящего формата файла – это компромисс между уровнем прозрачности, глубиной цвета и степенью сжатия. Вы должны осознавать, какие возможности имеются в вашем распоряжении.



Дизайнер **Дэвид Хелльсинг** (David Hellsing)
www.csszengarden.com/102

Революция!

Включение изображений в таблицу стилей

Графика играет важную роль для передачи настроения, задания тональности и привлечения взгляда зрителя. Не менее важны правильные способы включения графики в состав страницы.

Как первая автоматная очередь на тихом до той поры поле боя, доминантный образ сразу же привлекает внимание. Осознавая выразительную силу начальной иллюстрации, Дэвид Хелльсинг обратился к старым пропагандистским плакатам и таким приземленным источникам изображений, как намокшая бумага, чтобы задать нужную ему тональность. Кровь и краска, грязь и звезды – составные части «Революции» – гарантируют пронизывающий, берущий за душу визуальный эффект.

Ключом к дизайну было соблюдение баланса: в противовес военным и политическим ассоциациям, вызываемым заголовком, Хелльсинг выбрал спокойное и легкое оформление области контента. Для создания объектов и типографских эффектов он воспользовался программой Adobe Illustrator, а затем объединил все вместе в Adobe Photoshop.



Рис. 1 ✦ Фоновое изображение с видимым текстом, примененное к элементу h1

СОВЕТ

Обратите внимание, что порядок следования атрибутов фонового изображения существенен; например, для позиционирования нужно, чтобы сначала указывалась позиция по горизонтали, а потом по вертикали, то есть left, затем top. Впрочем, когда позиция описывается ключевыми словами (как, скажем, left и top), это не очень важно, а вот если применяются единицы измерения, к примеру, % или px, то неправильный порядок приводит к ошибке позиционирования.



Рис. 2 ✦ Фоновое изображение обрезано по вертикали в том месте, где кончается текст

Практическое применение

Разрабатывая дизайн страницы с помощью CSS, нужно придумать какой-то способ включения таких хорошо проработанных изображений, как заголовок в работе Хелльсинга. Дизайнера, привыкшего к тегу img, знакомство с исходным текстом сайта Zen Garden приведет в замешательство, поскольку механизм вставки графики неочевиден. Понятно, что вся работа возложена на CSS-файл, но как именно это делается?

Есть несколько способов связать картинку с элементом с помощью таблицы стилей, но лишь один из них достаточно хорошо поддерживается на практике – фоновые изображения. Чтобы присоединить картинку, скажем, к тегу h1 в Zen Garden (рис. 1), применяется довольно простой синтаксис: нужно задать путь к файлу картинки, его позицию и способ повтора (если его вообще нужно повторять):

```
h1 {
 background: url(path/to/image.gif) left top
no-repeat;
}
```

Если задан атрибут no-repeat, то, скорее всего, для изображения необходим объемлющий элемент с точно заданными размерами, за который оно не должно выступать. Если не заданы ширина width или высота height, то (если, конечно, изображение не совсем крохотное) высоты текста окажется недостаточно, чтобы вместить всю картинку, в результате чего она обрежется (рис. 2). В большинстве случаев следует указывать размеры исходного элемента, достаточные для всего фонового изображения:

```
h1 {
 background: url(path/to/image.gif) left top
no-repeat;
 width: 400px;
 height: 20px;
}
```

Подстановка изображения

Не всегда желательно оставлять текст поверх фонового изображения. Во многих вариантах дизайна Zen Garden изображения подставляются вместо различных заголовков и скрывают текст.

В марте 2003 года Дуглас Боумен опубликовал на своем сайте Stopdesign статью (www.stopdesign.com/articles/replace_text), в которой описывалась техника замещения текста фоновым изображением ради улучшения типографских черт страницы. Позже этот метод получил название «подстановка изображения Фарнера» (Fahrner Image Replacement – FIR) в честь Тода Фарнера (Todd Fahrner), который первым предложил его. Основная идея проста: поместите текст внутрь элемента span, скройте содержимое этого span и свяжите изображение с самим элементом. Вот пример подобной разметки:

```
<h1 id="pageHeader"><span>css Zen Garden</span></h1>
```

Теперь для подстановки изображения достаточно применить такую таблицу стилей:

```
#pageHeader {
  background: url(lemonfresh.gif) top left no-repeat;
  width: 400px;
  height: 20px;
}
#pageHeader span {
  display: none;
}
```

Любой span внутри элемента #pageHeader будет полностью скрыт от браузера, если применить к нему свойство display: none или visibility: hidden. Хелльсинг попробовал и то и другое – с одинаковым результатом. Этот прием очень популярен, не будь его, сам сайт Zen Garden вряд ли состоялся бы. Подстановка изображений играет фундаментальную роль в обеспечении максимальной гибкости дизайна на основе CSS.

Чем грозит подстановка изображений

В оригинальном варианте техники подстановки изображений, в котором использовалось свойство display: none, ставилась задача не просто заменить текст, но сделать это разумно. Компьютер не может прочитать текст, нарисованный на картинке, если ему немного не помочь. Если бы картинка не сопровождалась альтернативным описанием с помощью атрибута alt, то элемент img был бы вообще бесполезен для поисковых систем типа Google и для альтернативных программ представления страницы, не способных прорисовывать изображения (например, программ чтения с экрана, которые воспроизводят содержимое экрана голосом). Но, поскольку FIR помещает изображение поверх текста, уже имеющегося в HTML-документе, то все должно бы работать.

ПРИМЕЧАНИЕ

Включить изображение можно также с помощью сгенерированного контента. Этот прием позволяет заменить имеющийся контент элемента своим собственным:

```
h1 {
  content: url(aldente.gif);
}
```

Первоначально (в CSS2) это был способ манипулирования псевдоэлементами :before и :after, но позднее – в CSS3 – превратился в гораздо более универсальное свойство. Однако браузеры только-только начинают поддерживать его, так что пока мы рекомендуем пользоваться сгенерированным контентом только ради эксперимента.

> **ПРИМЕЧАНИЕ**
>
> Программа чтения с экрана пользуется теми же синтаксическими анализаторами, что и основные браузеры, например Internet Explorer, поэтому любой текст, не прорисовываемый браузером, не будет и произноситься.

Однако выявилась проблема. Google может прочитать скрытый текст, а вот некоторые программы чтения с экрана не могут. Схема FIR задумывалась для того, чтобы предоставить доступ к описанию изображения незрячим пользователям, но в некоторых случаях она просто не работает. Кроме того, если пользователь отключает показ графики, но оставляет CSS включенным, то браузер вообще ничего не покажет. Ну и, наконец, дополнительные span тоже что-то «весят», это, конечно, небольшая проблема, но все-таки...

Улучшенная методика

Когда описанный выше метод FIR приобрел популярность, упомянутые проблемы вышли на передний план. Может ли дизайнер со спокойной совестью применять технику, которая скрывает содержимое некоторых элементов от части людей и программ? К счастью, примерно в то же время начали появляться альтернативные методики подстановки изображений. Во многих из них какие-то фундаментальные проблемы решались, тогда как другие оставались нерешенными. В некоторых применялись тяжеловесные трюки с целью обеспечить межбраузерную совместимость. Когда писалась эта книга, еще не было найдено идеальной во всех отношениях схемы; имелись только более или менее удачные компромиссы.

> **ПРИМЕЧАНИЕ**
>
> Были высказаны частные мнения (например, статья Дэйва Ши «In Defense of Fahrner Image Replacement», www.digital-web.com/articles/in_defense_of_fahrner_image_replacement) и проведены исследования (см. статью Джо Кларка «Facts and Opinions About Fahrner Image Replacement», www.alistapart.com/articles/fir). В результате стало очевидно, что схема FIR в своем первоначальном виде неработоспособна – фундаментальный метод, так много сделавший для признания CSS, оказался неприменим на практике. И что теперь?

Схема Леа-Лэнгбриджа

Симус Леа (Seamus Leahy) и Стюарт Лэнгбридж (Stuart Langbridge) независимо открыли метод, позволяющий избавиться от лишних span и визуально скрыть текст, оставив его в то же время доступным программам чтения с экрана. К сожалению, из-за ошибок в реализации модели прямоугольных областей в версиях Internet Explorer для Windows младше 6.0 пришлось придумывать специальные обходные пути для этих браузеров.

> **ПРИМЕЧАНИЕ**
>
> На личном сайте Дэйва Ши есть страница, где рассматриваются различные вариации на тему подстановки изображений (www.mezzoblue.com/tests/ revised-image-replacement).

Разметка

```
<h3 id="header">I like cola.</h3>
```

Таблица стилей

```
#header {
  padding: 25px 0 0 0;
  overflow: hidden;
  background-image: url(cola.gif);
  background-repeat: no-repeat;
  height: 0px !important;

  height: /**/:25px;
}
```

Аргументы за: доступна программам чтения с экрана; нет лишних span.

Против: не решает проблему отключения графики при включенном CSS. Необходимы обходные трюки.

Поддерживается браузерами: Windows – Internet Explorer 5.0+, Netscape 7, Opera 6+, Firefox. Macintosh – Internet Explorer 5.2, Safari, Firefox.

Схема Рандла

Дизайнер Майк Рандл (Mike Rundle) предложил решение, в котором для выталкивания текста за левую границу окна используется отрицательное значение свойства text-indent. Решение простое и элегантное, хотя в некоторые случаях неприменимое, так как Internet Explorer для Windows 5.0 вместе с текстом выталкивает фоновое изображение.

Разметка

```
<h3 id="header">Apple pie with cheddar?!</h3>
```

Таблица стилей

```
#header {
 text-indent: -5000px;
 background: url(sample-image.gif) no-repeat;
 height: 25px;
}
```

Аргументы за: доступна программам чтения с экрана; нет лишних span. Красивая и несложная таблица стилей.

Против: не решает проблему отключения графики при включенном CSS. Не всегда правильно работает в Internet Explorer 5.0 для Windows.

Поддерживается браузерами: Windows – Internet Explorer 5.5+, Netscape 7, Opera 6+, Firefox. Macintosh – Internet Explorer 5.2, Safari, Firefox.

Схема Левина

Александр Левин выступил с парадоксальной идеей: вместо того чтобы размещать текст внутри span, вынесем его наружу и оставим то и другое внутри родительского элемента, а затем уже пустым span накроем текст. Если все сделать правильно, то программы чтения с экрана будут довольны, и проблема выключенной графики при включенном CSS тоже будет решена. Правда, возникает новая трудность: изображение

не должно быть прозрачным, иначе сквозь него будут видны фрагменты текста. Да и соответствующая таблица стилей – зрелище не для слабонервных.

Разметка

```
<h3 id="replace" id="myh1">And a dash of
Thyme.<span></span></h1>
```

Таблица стилей

```
.replace {
 position: relative;
 margin: 0px; padding: 0px;
 /* скрыть overflow:hidden от IE/Mac \*/
 overflow:hidden;
 /* */
}
.replace span {
 display: block;
 position: absolute;
 top: 0px;
 left: 0px;
 z-index: 1; /* для Opera 5 и 6 */
}
#myh1, #myh1 span {
 height: 25px;
 width: 300px;
 background-image: url(thyme.png);
}
```

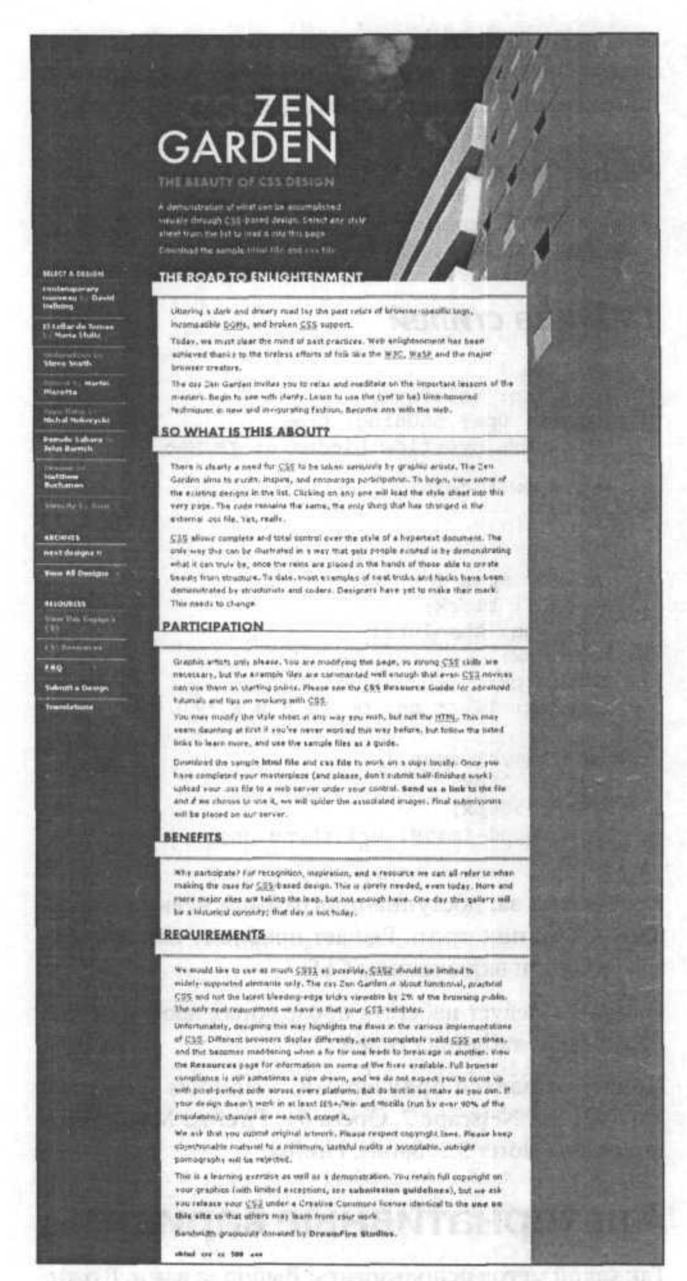
Аргументы за: доступна программам чтения с экрана; нет лишних span. Решает проблему отключения графики при включенном CSS.

Против: Следует избегать прозрачных изображений. Громоздкая таблица стилей.

Поддерживается браузерами: Windows – Internet Explorer 5+, Netscape 7, Opera 6+, Firefox. Macintosh – Internet Explorer 5.2, Safari, Firefox.

Альтернативные варианты

Так какой метод использовать? Выбор за вами. В ожидании того дня, когда большинство браузеров начнут поддерживать автоматически сгенерированный контент, методы Леа–Ленгбриджа, Рундла и Левина – это лучшее, на что можно рассчитывать. Существующие варианты дизайна сайта Zen Garden, в том числе и «Революция» Хелльсинга были бы невозможны без того прогресса, к которому привела оригинальная схема FIR; в будущих вариантах, наверное, станут применяться альтернативные схемы, обеспечивающие большую доступность.



Дизайнер **Марк Трудель** (Marc Trudel)
www.csszengarden.com/094

Арт Деко

Минимум графики, максимум воздействия

Приступая к работе над новым дизайном, Марк Трудель следует некоему ритуалу. Сначала он заходит на сайты, которые вдохновляют его, примерно час рассматривает чужие работы и подбирает контент: фотографии, иллюстрации, текст. Затем открывает Adobe Photoshop, где готовит миникартинки (thumbnails). В отличие от многих художников-графиков, он не любит бумажных эскизов, считая, что им не хватает уровня детализации, достижимого в Photoshop.

В начале зимы 2004 года Трудель нарушил этот ритуал, поскольку в то время гулял по улицам Майами. Решив, что здания в районе Саут Бич, выстроенные в стиле Арт Деко, – прекрасная модель для абстрактной фотографии крупным планом, Трудель сделал множество снимков, один из которых лег в основу чистых и свежих линий работы, представленной на сайт CSS Zen Garden.

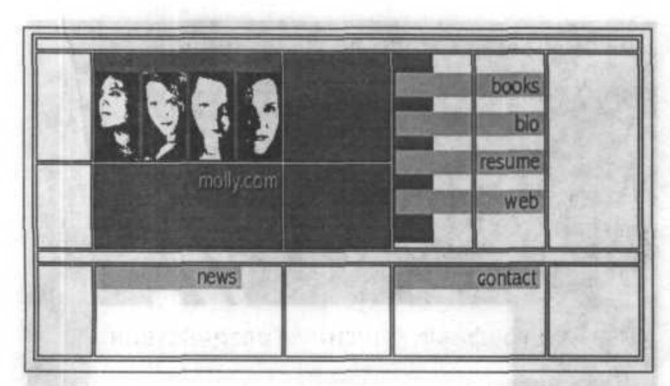


Рис. 1 ✦ Ограничительная сетка, присущая табличной верстке



Рис. 2 ✦ Дизайн «Арт Деко» – открытый и воздушный. Хотя с помощью таблиц сверстать такую страницу тоже можно, но, скорее всего, раньше это никому не пришло бы в голову

Новый образ мышления

Когда дело доходит до работы с графикой в контексте CSS, от дизайнеров «старой школы», привыкших составлять веб-страницу из кусочков, требуется научиться думать по-новому. Для читателей, никогда, по счастью, не сталкивавшихся с методом «мозаики», поясним, что в этом случае дизайнер готовит макет в Photoshop, затем разрезает картинки на кусочки и помещает их в HTML-таблицы, которые образуют на странице сетку. Такой подход естественно преобладал в веб-дизайне на протяжении большей части 1990-х годов (рис. 1). Увы, эта практика не отмерла и сегодня, что тормозит прогресс и наносит ущерб доступности и функциональности.

Один из редко обсуждаемых недостатков метода мозаики – это его ограниченные возможности, в особенности относительно графики. Нужно, чтобы каждый кусочек изображения точно укладывался в ячейку сетки, представляющей собой набор прямоугольников. Мягко говоря, это мешает дизайнеру в полной мере раскрыть свой творческий потенциал. Поэтому у многих и многих дизайнеров вошло в привычку создавать излишне геометричные макеты.

Применение CSS в качестве инструмента верстки снимает многие подобные ограничения. Конечно, и в CSS все основано на модели прямоугольных областей, от этого никуда не уйти, но есть и различия. Работая с таблицами стилей, вы можете мыслить в терминах сетки, а можете и забыть о ней. Проявив изобретательность, дизайнер может вырваться за пределы сетки и предложить более текучий, открытый дизайн (рис. 2).

Не будучи связан таблицами, дизайнер может подойти к холсту, не думая заранее об ограничениях. И благодаря этому изображения теперь используются совсем иначе, чем раньше, что дает прекрасные результаты.

Жесткие сетки, так затруднявшие работу дизайнера, остались в прошлом. CSS позволяет прикрепить изображение к любому элементу в качестве фона, что, в свою очередь, дает возможность организовывать слои, перекрытие и – при наличии хорошего браузера – даже пользоваться прозрачностью формата PNG для создания целого ряда визуальных эффектов.

Технология CSS не только позволила дизайнеру раскрыть свой творческий потенциал, но и предложила способы уменьшения веса страниц, к вящему удовольствию пользователей.

Разумные подходы к работе с графикой

Чистые, открытые линии «Арт Деко» учат нас, в частности, тому, как следует подходить к работе с графикой при использовании таблиц стилей. CSS позволяет иначе взглянуть на создание и размещение изображений.

Размер и вес изображения

Картинки должны весить как можно меньше, но не в ущерб зрительной целостности. Метод разрезания, применявшийся в эпоху табличной верстки, способствовал, в частности, сокращению общего веса страницы.

Однако таблицы стилей позволяют разместить понравившееся вам изображение точно в нужном месте. В некоторых случаях мы берем более крупную картинку, в других – размером поменьше, а иногда выстилаем большую область совсем крохотными изображениями.

Взгляните на основную фотографию, образующую фон «Арт Деко». На ней изображены облака и характерные формы здания. Именно эта часть дизайна создает впечатление легкости и воздушности, но, рассмотрев фоновую картинку в деталях, мы будем немало удивлены размерами (рис. 3).

Размер изображения составляет 788×497 пикселей. Представители старой школы разрезали бы его на несколько частей (если дизайнеру вообще пришла бы в голову мысль так оформить страницу).

Все изображение весит 45,02 Кб; в табличной верстке считается, что это слишком много. Но напомним, что CSS предлагает другие подходы к использованию графики, поэтому такой вес основного фона не покажется чрезмерным, если прочих картинок мало и размер их невелик. Следующим по «массивности» идет изображение с заголовком, которое весит всего 3,28 Кб и имеет более привычный размер 266×103 пикселя (рис. 4).

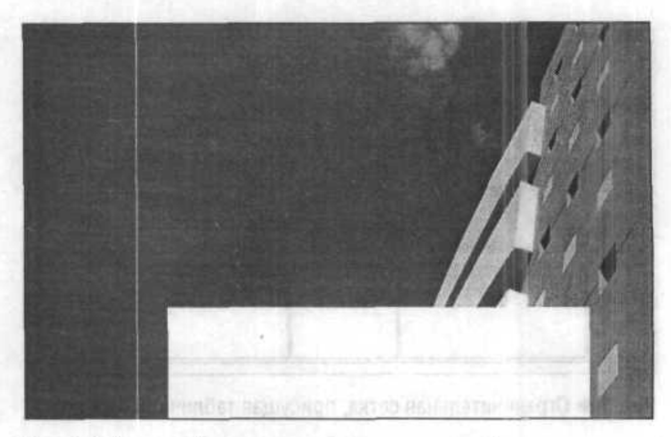


Рис. 3 ✦ Главное фоновое изображение в «Арт Деко»



Рис. 4 ✦ Прозрачный GIF, взятый в качестве картинки с заголовком

Число изображений

В дизайне на основе CSS число изображений не обязательно должно быть меньше, чем в традиционном табличном макете, но и с меньшим количеством вы можете достичь более впечатляющих результатов. В табл. 1 описаны все картинки, использованные в «Арт Деко» с указанием размера и веса.

Таблица 1 ✦ Использование картинок в «Арт Деко»

Картинка	Размеры (в пикселях)	Размер файла	Применение
bg.jpg	788×497	45,02 Кб	Основной фон, фотография
bg_blue1.gif	2000×1	0,14 Кб	Фон
csszengarden.gif	266×1033	28 Кб	Заголовок страницы
thebeauty.gif	263×13	1,69 Кб	Подзаголовок страницы
h1.gif	537×39	3,02 Кб	Заголовок уровня 1
h2.gif	537×50	2,83 Кб	Заголовок уровня 2
h3.gif	537×50	2,46 Кб	Заголовок уровня 3
h4.gif	537×50	2,20 Кб	Заголовок уровня 4
h5.gif	537×50	2,52 Кб	Заголовок уровня 5
lightblue.gif	125×3	0,06 Кб	Фон списка ссылок
selectdesign.gif	95×7	0,37 Кб	Заголовок Select design
archives.gif	95×7	0,28 Кб	Заголовок Archives
resources.gif	95×7	0,31 Кб	Заголовок Resources

Так как же уменьшить число используемых изображений и сократить тем самым вес страницы? Одна из целей данной главы – побудить вас мыслить о графике более широко, и ответ на этот вопрос будет неплохим началом. Есть несколько способов сократить потребность в графике, а именно:

- всюду, где возможно, пользоваться стилизованным текстом вместо картинок. В «Арт Деко» так делается в заголовке и подзаголовке страницы. Для получения типографских эффектов действительно применена графика, но непосредственно под ней расположен обычный текст (из раздела `quickSummary`). Соблюдение такого рода баланса может оказаться очень полезным;

ПРИМЕЧАНИЕ

Конечно, современные утилиты для сжатия графики позволяют получать файлы меньшего размера. В «Арт Деко» вес основного изображения можно было бы значительно уменьшить без заметной потери качества. Но поскольку эта фотография является центральным элементом дизайна и так зримо доминирует в визуальном пространстве, то автор имеет право пожертвовать весом ради достижения более яркого впечатления.

- брать только ту часть картинки, которая необходима для достижения желаемого эффекта. Нагляднее всего это проявляется в использовании основной фоновой фотографии и заголовков, которые и придают всей странице зрительное очарование;
- пользоваться очень тонкими изображениями для заполнения фона. Фон части документа, расположенной под основной фотографией, залит картинкой bg_blue1.gif, высота которой всего 1 пиксель, а весит она жалкие 140 байтов;
- по возможности используйте изображения повторно. В умело построенной таблице стилей одно и то же изображение можно связать со многими элементами. В «Арт Деко» картинка lightblue.gif служит фоном для трех разных элементов, но при этом задано только одно правило, которое с помощью селектора потомков применяется ко всем элементам ul внутри раздела linkList:

```
#linkList ul {
  margin: 0px;
  padding: 0px;
  background-image: url(lightblue.gif);
  background-repeat: repeat-y;
  }
```

- не забывайте об обычной заливке цветом. Если нужного эффекта можно добиться с помощью свойства color или background-color, не пользуйтесь изображениями.

И наконец, если картинка лишняя, выбросьте ее. У дизайнера всегда возникает сильное искушение вставить избыточную графику. Спросите себя, каждая ли картинка на странице имеет какое-то назначение, эстетическое или функциональное. Если ответ неясен, подумайте, не стоит ли от этой картинки избавиться.

Слои графики

Еще один вопрос, которым редко задавались до реализации поддержки CSS в браузерах, – это организация нескольких слоев графики для создания тех или иных эффектов. Сегодня это вошло в плоть и кровь дизайнеров, но раньше размещение одного изображения поверх другого практиковалось редко.

Встречался такой метод разве что при использовании прозрачного GIF поверх фона или цвета, указанного

Рис. 5 ✦ Наложение прозрачного изображения на фон, в духе старой школы

для body или table. В этом случае в теге body задавалось фоновое изображение или цвет для всей страницы, например:

```
<body bgcolor="#3366CC" background="image/
bg.jpg">
```

Затем где-то внутри тела страницы, чаще всего в ячейке таблицы, размещалась прозрачная картинка:

```
<td><img src="images/csszengarden.gif"
width="266" heigh="103"
alt="CSS Zen Garden"></td>
```

В результате получалось что-то похожее на фрагмент рассматриваемого дизайна, изображенный на рис. 5.

В «Арт Деко» Трудель использовал основную фотографию как фон для элемента #container:

```
#container {
  background: url(bg.jpg) no-repeat scroll
  left top;
}
```

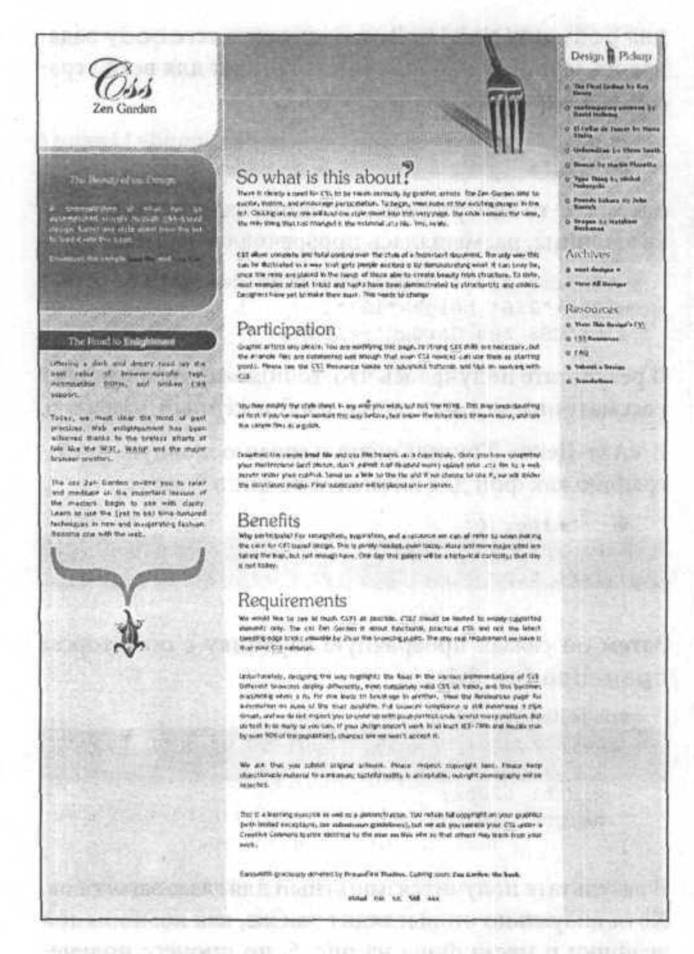
Затем он связал прозрачную картинку с селектором #pageHeader h1:

```
#pageHeader h1 {
  background: url(csszengarden.gif) no-repeat
  left top;
  width: 620px;
  height: 103px;
}
```

В результате получится приятный для глаза заголовок. Хотя визуально он выглядит так же, как комбинация графики и цвета фона на рис. 5, но процесс получения эффекта совсем иной.

Максимальное воздействие

В «Арт Деко» для создания свежего решения используется не очень много графики. Добавьте к этому изобретательное размещение картинок, тщательно продуманный баланс между типографскими эффектами, достигаемыми за счет графики, и текстом, а также простое, но повсеместное применение изображений и текста для обеспечения нужной цветности, – и вы получите дизайн, в равной мере технически оснащенный, впечатляющий и не ограничивающий творческую свободу автора.



Дизайнер **Михал Мокржицки** (Michal Mokrzycki)
www.csszengarden.com/097

Без границ!

**Долой прямоугольники,
да здравствуют скругленные изображения!**

Преобладающая тенденция в дизайне на основе CSS – это центрированная, фиксированная верстка и предсказуемые тени. Именно от этой тенденции Михал Мокржицки решил отойти в своей работе «Без границ». Он остановился на расширяемом трехколонном макете во всю ширину окна. Дизайн должен был быть максимально простым, но в то же время элегантным и своеобразным.

В отличие от многих дизайнеров, строго следующих процедуре, Мокржицки часто полагается на импровизацию. Начал он с просмотра своей коллекции фотографий, а потом переходил от Adobe Photoshop к CSS и обратно, стремясь воплотить возникшие идеи в жизнь. Работая над этим дизайном, Мокржицки осознал, что CSS предлагает безграничные возможности для выхода за пределы «прямоугольной модели», потому-то он и назвал свое творение «Без границ».

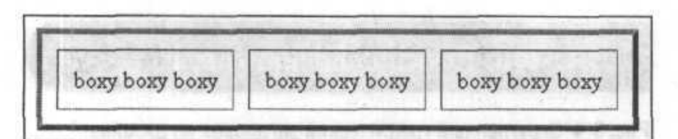


Рис. 1 ✦ Избыток прямоугольников на ранних стадиях развития Web

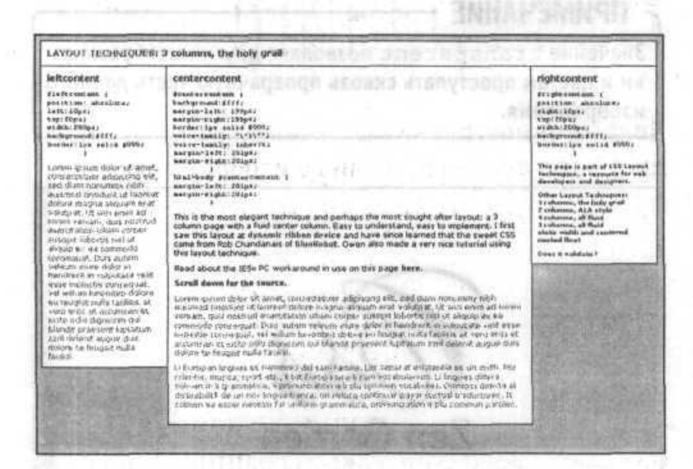


Рис. 2 ✦ Пример ранней верстки на основе CSS: действительно, прямоугольников слишком много, но для своего времени это было новаторством, поскольку зрительно воспроизводилась популярная на тот момент HTML-верстка

ПРИМЕЧАНИЕ

К счастью, по мере развития технологий веб-дизайнеры научились придавать форму табличному макету методом «мозаики»: когда картинки режутся на части, помещаемые затем в ячейки таблицы. Со временем был выработан и более процедурно-ориентированный подход, приведший к новым визуальным тенденциям, но «прямоугольность» все же присутствовала, поскольку в Web не было другого графического интерфейса.

Проблема прямоугольников

Совсем неслучайно дизайн для Web исторически сложился «прямоугольным». И какая ирония заключена в том, что именно дизайн на основе CSS особенно часто обвиняли в этом грехе! Притом именно табличные макеты, для которых характерно активное применение границ и отступов, часто выглядели как нагромождение прямоугольников, бессистемно разбросанных по поверхности страницы (рис. 1).

Еще вспомните, что в те времена большинство персональных компьютеров оснащались мониторами с разрешением 640×480 пикселей – совсем немного по сравнению с тем, что мы имеем сейчас. Добавьте к этому основанный на прямоугольниках интерфейс самого веб-браузера и форму компьютерного экрана. Вот и получится прямоугольник внутри прямоугольника внутри прямоугольника.

С визуальной точки зрения посетителю было не очень удобно разбираться в информации, глубоко упрятанной в скоплении прямоугольных областей. Но то была эпоха головокружения – Всемирная паутина только появилась, и очарование ее было столь велико, что и создатели страниц, и зрители готовы были простить многое.

Разумеется, когда для создания визуального представления стали применять технологию CSS, прежде всего попытались на новой основе воссоздать старые, уже привычные типы верстки. В ранних макетах преобладала структура с несколькими колонками, к которой легко было применять декоративные элементы вроде границ. Именно поэтому на ранние работы обрушился град критики за изобилие прямоугольных форм (рис. 2).

Неправильно было бы сказать, что CSS несет основную долю ответственности за «проблему прямоугольников». По мере того как наше понимание CSS становится более зрелым, применяются все новые и новые способы работы с графикой, особенно с фоновыми изображениями для элементов, для создания многослойных форм, орнаментов и других украшений, позволяющих выйти за границы прямоугольных макетов прошлых лет.

В дизайне «Без границ» применяются скругленные углы и круговые элементы декора, позволяющие убрать угловатость, присущую прямоугольной верстке, и смягчить общее впечатление от страницы.

Да здравствуют изгибы!

В работе «Без границ» есть много примеров того, как можно добавить изгибы и скруглить формы, чтобы они не казались слишком прямоугольными. Для этого так или иначе используются фоновые изображения.

Простые изгибы: одно фоновое изображение

Самый очевидный способ нарушить «прямизну» – создать единственную картинку, содержащую все кривые и контент. Затем она связывается с некоторым элементом посредством таблицы стилей.

«Без границ» демонстрирует применение такого метода. У заголовка The Road to Enlightenment приятно скругленный правый край (рис. 3).

Затем это изображение было присоединено к элементу h3 в разделе #preamble:

```
#preamble h3 {
  background: transparent rl(enlightenment.gif)
 no-repeat top left;
  width: 291px;
  height: 37px;
  margin-bottom: 10px;
  padding: 0;
}
```

Но что делать, если вы хотите наложить такую непрямоугольную картинку на другой фон?

Составное фоновое изображение

Сделать страницу более интересной можно, воспользовавшись фоновыми изображениями, присоединенными к нескольким элементам. При этом создается иллюзия многослойности, а картинки остаются небольшими по размеру и весу.

В дизайне «Без границ» такая техника применяется в заголовке CSS Zen Garden, у которого скруглен правый нижний угол, а под ним открывается фон с диагональными полосками (рис. 4).

Рис. 3 ✦ Скругленный графический заголовок из дизайна «Без границ». Обратите внимание на прозрачность правого края

ПРИМЕЧАНИЕ

Значение `transparent` позволяет другим изображениям и цветам проступать сквозь прозрачную часть данного изображения.



Рис. 4 ✦ Заголовок со скругленным правым углом кажется наложенным на полосатый фон

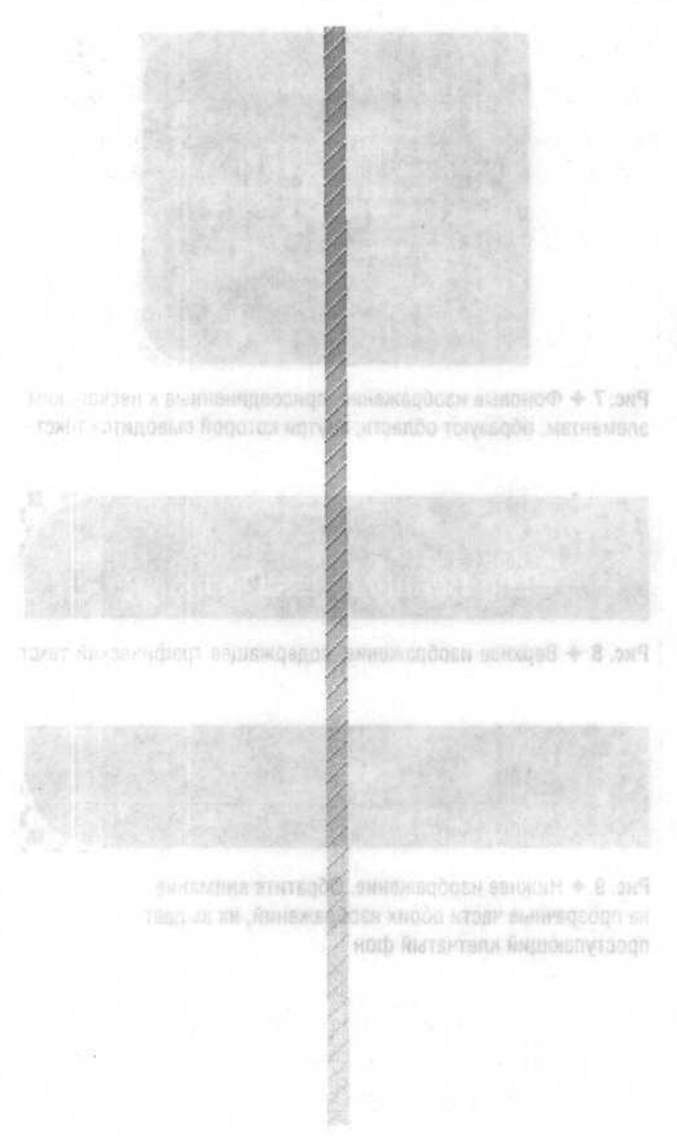


Рис. 5 ✦ Тонкая картинка повторяется вдоль оси x, образуя фон элемента body



Рис. 6 ✦ Еще один пример эффекта двухслойного фона

Чтобы достичь такого эффекта, Мокржицки сначала поместил на страницу картинку с полосками (рис. 5) в качестве фонового изображения, присоединенного к элементу body:

```
body {
 font: small "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
 color: #676767;
 margin: 0px;
 padding: 0px;
 background: #fbfbf2 url(page_back.gif)
repeat-x fixed;
 background: transparent url(body_back.png)
no-repeat top left;
}
```

Затем к элементу #container он присоединил изображение со скругленным углом:

```
#container {
 padding: 0px;
 margin: 0px;
 background: transparent url(body_back.png)
no-repeat top left;
}
```

Концептуально, два фоновых изображения теперь перекрывают друг друга, и выглядит это как прямоугольник со скругленным углом на фоне полосок. Другой пример такого рода можно наблюдать в правом верхнем углу страницы, где заголовок списка ссылок размещен на фоне тех же полосок (рис. 6).

Оформление углов области контента

Основываясь на идее слоев, можно скруглить несколько углов у области контента или добиться других эффектов. Но фокус в том, что для этой цели потребуется несколько элементов в HTML-разметке.

Примером могут служить разделы #pageHeader и #quickSummary в дизайне «Без границ». В результате текст оказывается внутри области со скругленными нижним и верхним углом (рис. 7).

Без применения CSS понадобилось бы одно или несколько изображений, упакованных в таблицу. CSS же позволяет создать только изображения для получения эффекта скругления и применить стили к HTML-тексту, уменьшив тем самым общий размер картинок.

Для этого Мокржицки создал два изображения: одно для верхней части (рис. 8) и другое – для нижней (рис. 9).

Затем они были присоединены в качестве фона к элементам h2 в разделах #pageHeader и #quickSummary соответственно:

```
#pageHeader {
 position: absolute;
 left: 0px;
 top: 190px;

 width: 291px;
 height: 159px;
 background: transparent url(green_top.gif)
no-repeat top left;
 z-index: 3;
}

#pageHeader h2 {
 position: absolute;
 top: 118px;
 background: transparent url(green_bottom.gif)
no-repeat bottom left;
 width: 291px;
 height: 135px;
 margin: 0;
 padding: 0;
}
```

Но с элементами #pageHeader h1 и #quickSummary не связано никаких изображений. Вместо этого они позиционированы, для элемента h1 задан цвет фона, а текст в HTML-разметке стилизован:

```
#pageHeader h1 {
 position: absolute;
 top: 59px;
 background-color: #bbbf58;
 border-right: 2px solid #fff;
 width: 289px;
 height: 150px;
 margin: 0;
 padding: 0;
}
#quickSummary {
 position: absolute;
 top: 270px;
 left: 20px;
 width: 245px;
 line-height: 2;
 z-index: 3;
}

#quickSummary p {
 color: #fff;
 font-size: 11px;
 line-height: 1.5;
 font-family: Verdana, Arial, Helvetica, sans-
serif;
}
```



Рис. 7 ✦ Фоновые изображения, присоединенные к нескольким элементам, образуют область, внутри которой выводится текст

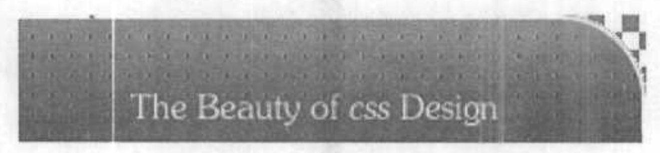


Рис. 8 ✦ Верхнее изображение, содержащее графический текст



Рис. 9 ✦ Нижнее изображение. Обратите внимание на прозрачные части обоих изображений, их выдает проступающий клетчатый фон

ПРИМЕЧАНИЕ

Идея многослойного фона лежит также в основе более сложных приемов, например, описанных в статьях Дугласа Боумена «Sliding Doors of CSS» (www.alistapart.com/articles/slidingdoors и www.alistapart.com/articles/slidingdoors2).

ПРИМЕЧАНИЕ

Много мыслей по поводу аналогичных подходов к получению различных угловых эффектов вы найдете в статье Серена Мадсена (Soren Madsen) «CSS Design: Creating Custom Corners and Borders» (www.alistapart.com/articles/customcorners).

- The Final Ending by Ray Henry
- contemporary nouveau by David Hellsing
- El Collar de Tomas by Maria Stultz
- OrderedZen by Steve Smith
- Bonsai by Martin Plazotta
- Type Thing by Michal Mokrzycki
- Pseudo Sahara by John Barrick
- Dragen by Matthew Buchanan

Archives

- next designs »
- View All Designs

Resources

- View This Design's CSS
- CSS Resources
- FAQ
- Submit a Design
- Translations

Рис. 10 ✦ Список ссылок с нестандартными маркерами

Обратите внимание на высоту, заданную для разделов. Это позволяет избежать перекрытия и переполнения.

Нестандартные маркеры

Этот прием также подразумевает присоединение фонового изображение к элементу, в данном случае к элементу li. Круглые графические маркеры хорошо укладываются в основную тему дизайна «Без границ» и дают пищу для размышлений тем дизайнерам, которым интересно, как создается графическое оформление, интегрированное с общей концепцией (рис. 10).

Для создания полного набора маркеров для своей работы Мокржицки сначала подготовил три разных изображения. Первое – светло-зеленое – нужно было для выделения основных ссылок; второе – голубое – для ссылок на архивы; третье – красновато-коричневое – для ссылок на ресурсы.

Затем изображения маркеров были присоединены в качестве фона к конкретным элементам:

```
#lselect li {
 background: transparent url(link_dot.gif) 0
7px no-repeat;
}

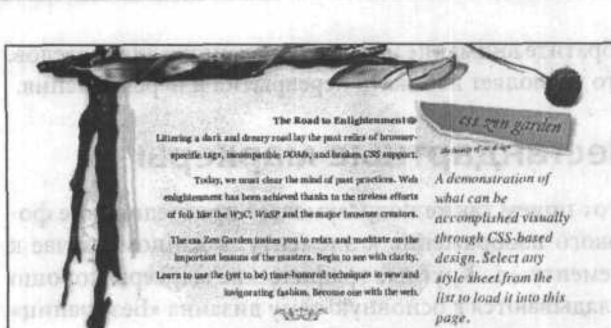
#larchives li {
 background: transparent url(link_dot2.gif) 0
7px no-repeat;
}

#lresources li {
 background: transparent url(link_dot3.gif) 0
7px no-repeat;
}
```

Разноцветные маркеры немного оживляют дизайн и дают понять посетителю, что семантически эти разделы различны.

Скругленные изображения

Когда вы пытаетесь создать скругленные углы или другие краевые эффекты, чтобы уйти от всем надоевших прямоугольных форм, техника присоединения изображений к элементам поможет придать странице свежий, неповторимый облик. А дополнительные формы сделают визуальные эффекты еще более выразительными, как в дизайне «Без границ».



css zen garden

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. Send us a link to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

Requirements

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the Resources page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see submission guidelines), but we ask you release your CSS under a Creative Commons license identical to the one on this site so that others may learn from your work.

Bandwidth graciously donated by DreamFire Studios.

xhtml css cc 508 aaa

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

select a design

gardenias de perfume by armando sosa
pneuma by adam polselli
birdhouse by justin viloria
defiance by james chiu
mediterranean by john whittet
austrian's dark side by rene grassegger
invitation by brad daily
odyssey by terrence conley

archives

next designs »
view all designs

resources

view this design's css
css resources
faq
submit a design
translations

Дизайнер **Дэйв Ши** (Dave Shea)
www.csszengarden.com/013

Береговой бриз

Подготовка графики для работы и поиск дополнительного исходного материала

Житель Ванкувера и известный писатель Дуглас Коупленд (Douglas Coupland) назвал свой родной город городом из стекла. Он привольно раскинулся у подножья горного хребта на западном побережье Канады. Навеянная морем тема с плавником и раковинами, дополненная несколькими осколками стекла, могла бы еще больше ассоциироваться с городом, если бы добавить несколько порций суши. Но дизайнер Дэйв Ши знал, где остановиться.

Графические образы, использованные в «Береговом бризе», были нарисованы вручную в Adobe Photoshop, никаких предварительно отснятых фотографий не было. Пользуясь различными инструментами, имеющимися в Photoshop, Ши сумел воспроизвести живописные текстурированные эффекты, не прибегая к фотографиям.

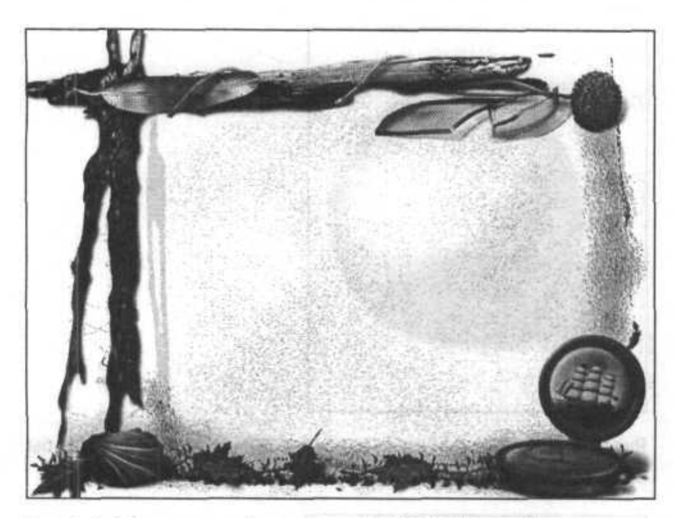


Рис. 1 ✦ Оригинальный, созданный в Photoshop файл, положенный в основу «Берегового бриза»



Рис. 2 ✦ Для осколков стекла создана прозрачная тень

Отправная точка

Основные объекты были когда-то созданы для давно уже почившего в бозе сайта, созданного задолго до Zen Garden. Вернувшись к оригинальному, созданному в 2001 году с помощью Photoshop файлу, состоящему из 300 с лишним слоев (рис. 1), Ши выбрал из него элементы, необходимые для нового дизайна. Адаптировать графику оказалось сравнительно просто; проблема состояла в том, чтобы создать макет, в который удачно вписался бы текст.

Исходный дизайн представлял собой почти квадратную область, размеченную с помощью HTML, а текст на сайте Zen Garden довольно длинный, поэтому верстка должна быть вытянута по вертикали. Предстояло как-то разрешить эту неувязку.

В качестве отправной точки у Ши была квадратная область. Оказалось, что приспособить ее к идее «Берегового бриза» проще всего, разрезав квадрат пополам и растащив верхнюю и нижнюю половины. L-образные уголки эффектно обрамляют страницу, оставляя необходимое пустое пространство.

Технические приемы

Конечно, не каждый будет рисовать весь дизайн вручную, но некоторые приемы, примененные в «Береговом бризе», могут оказаться полезными и в более широком контексте. Доработать существующее изображение или добавить визуальные штрихи проще, когда вы владеете необходимыми навыками. Описанные ниже приемы ориентированы на редактор Adobe Photoshop. Поскольку необходимые инструменты рисования есть во многих графических программах, то вы, скорее всего, сможете воспользоваться этими методами и в своем любимом редакторе.

Использование кистей и преобразований

Осколки стекла в правом верхнем углу страницы отбрасывают тень и кажутся прозрачными, как будто сквозь них просвечивает солнце (рис. 2). Чтобы достичь такого эффекта, нужно несколько больше, чем просто наложить имеющийся в Photoshop теневой

слой. Впрочем, ненамного больше. Начнем с дублирования образа на переднем плане (выберите из меню **Layer → Duplicate Layer**), в данном случае твердого стекла. Расположим новый слой под исходным и зальем сплошным цветом. С помощью команды **Free Transform** (**Edit → Free Transform**), позволяющей растягивать, сжимать, поворачивать и перекашивать геометрические фигуры, тень была сжата по вертикали, скошена влево, после чего размыта, чтобы сгладить внешние края (рис. 3).

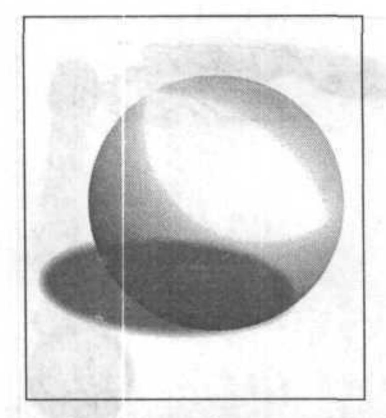


Рис. 3 ✦ Исходная форма со сплошной тенью, скошенной, чтобы создать иллюзию, будто она отбрасывается трехмерным телом

Чтобы сделать тень прозрачной, потребуются инструменты Eraser (ластик) и Smudge (растушевка). Тот и другой можно настроить, сделав больше или меньше, с мягкими или твердыми краями. Чтобы вручную стереть среднюю часть тени, была взята крупная кисть с мягкими краями; затем к результату была применена небольшая растушевка тоже с мягкими краями, чтобы сделать тень слегка неправильной (рис. 4). Идеально правильная тень, конечно, выглядит красиво, но в жизни объекты редко бывают безупречными, так что без дополнительной проработки деталей они будут казаться нереалистичными. Для завершения эффекта пришлось еще слегка подпортить поверхность исходного объекта (рис. 5).

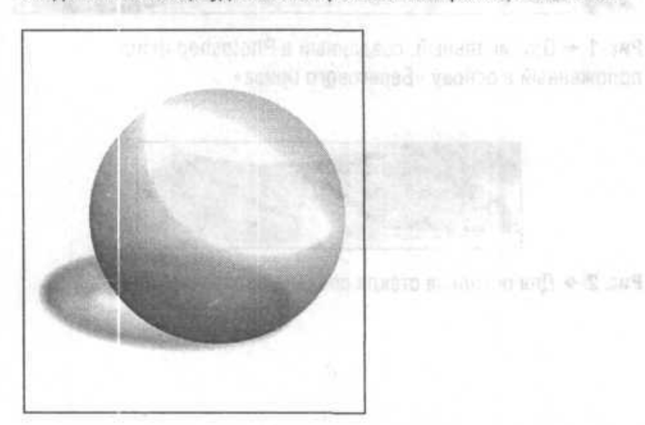


Рис. 4 ✦ Отбрасываемая объектом тень сделана прозрачной с помощью ластика и растушевки

Использование инструментов Burn и Dodge

Для реализации многих штришков можно было бы обойтись только ластиком и растушевкой, но вряд ли вы захотите ими ограничиться. Добавить тень или выделение к любому графическому изображению не очень трудно благодаря соответственно инструментам Burn (затемнитель) и Dodge (осветлитель). В качестве основного слоя был взят лист, залитый одним цветом (рис. 6), с помощью осветлителя были нанесены светлые прожилки. Дополнительную глубину им может придать затемнитель, с помощью которого накладываются легкие тени.

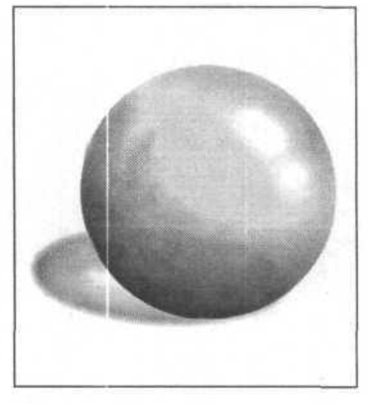


Рис. 5 ✦ Окончательный вариант с дефектами, добавленными для пущего реализма

Сетка прожилок на поверхности однотонного листа выглядит неэстетично, поэтому с помощью затемнителя привнесены дополнительные оттенки, чтобы подчеркнуть трехмерность (рис. 7).

Вышеупомянутые инструменты использовались для создания листьев лозы. Достаточно сделать копии исходного листа, а затем повернуть их и изменить размеры. Добавив саму ветвь и еще немного поработав над выделением и тенью, мы добились полной иллюзии (рис. 8).

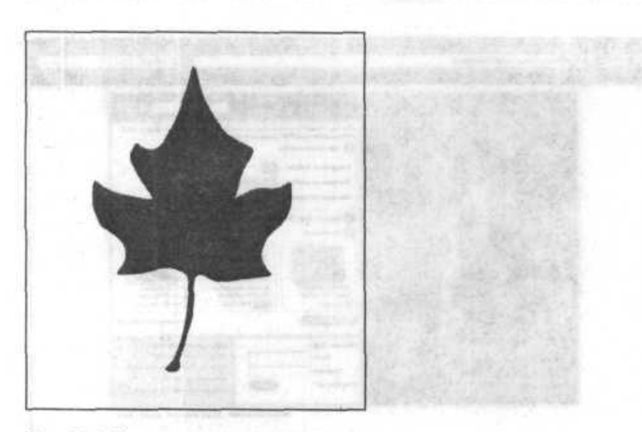


Рис. 6 ✦ Исходная поверхность листа



Рис. 7 ✦ Для имитации прожилок и текстуры добавлены тени и выделения

ПРИМЕЧАНИЕ

Цветной вариант рис. 6 и 7 см. на вкладке.

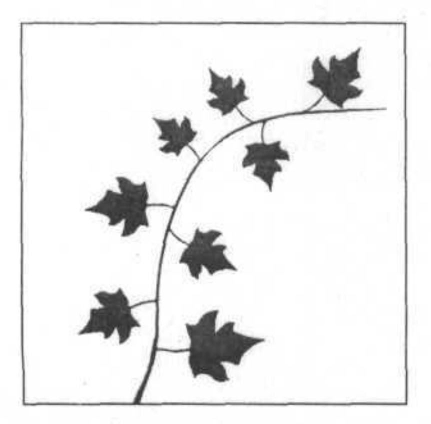


Рис. 8 ✦ Лоза вместе с листьями

СОВЕТ

Рекомендуем книгу Peter K. Burian «Mastering Digital Photography and Imaging» (Sybex, 2004).

Возможно, вы заметили, что по мере уменьшения размера листьев прожилки пропадают; когда масштаб изображения уменьшается, детали становятся неразличимыми. Но важно отметить, что глаз все же распознает многие мелочи, а общее впечатление усиливается даже от незаметных, на первый взгляд, визуальных ключей. Бывает очень обидно уменьшать масштаб хорошо и детально проделанной работы, возникает естественное желание оставить ее такой, как есть.

Создание графических образов

Хотя не каждый может позволить себе потратить деньги или время (а это много часов) на прорисовку изображений вручную, как в «Береговом бризе», тем не менее это один из способов подобрать картинки, необходимые для дизайна.

Обращение к традиционным графическим искусствам – это, быть может, самый очевидный способ создания изображений. Хотя рисование и живопись нечасто применяются в процессе разработки целого сайта, но многие дизайнеры сначала делают эскиз идеи на бумаге и только потом опускаются до уровня пикселей. Иметь под рукой архив рисунков, фотографий и иных материалов может быть полезно, когда вы устанете от обычных источников. В конце концов, в будущем вы сможете почерпнуть из них вдохновение, если они уж больше ни на что не пригодятся.

Таланты бывают разные, в наши дни дизайном для Web занимаются люди с различным образованием и навыками. Если вы не умеете рисовать, то, скорее всего, у вас есть другие возможности для добывания нужных картинок. Благодаря развитию технологии (и падению цен), с помощью дешевого сканера и цифровой фотокамеры можно получить высококачественный исходный материал, если только вы знаете, как этими устройствами эффективно пользоваться. Возможно, к устройству даже прилагалось руководство, но если вы хотите действительно познакомиться с основами, то имеет смысл почитать какое-нибудь введение в цифровую графику.

Покупка изображений

Конечно, ограниченность бюджетов, выделяемых на дизайн большинства сайтов, не позволяет платить за изображения, сделанные на заказ, хотя клиенты с толстым кошельком и могут оценить, насколько дополнительное внимание к теме и композиции окупается.

Работа на заказ и приобретение без отчислений от продаж

Чтобы найти местного фотографа или иллюстратора, придется покопаться в «желтых страницах», но часто лучших результатов можно достичь, общаясь с другими местными творческими личностями по сети. Впрочем, нет нужды ограничиваться только местными талантами, существует множество сайтов, на которых клиенты могут договориться о работе с профессионалами. Один из самых известных – это Aquent (www.aquent.com), но есть и немало других, работающих более или менее эффективно.

Работа на заказ стоит дорого, и не много найдется сайтов, которые могут себе это позволить. Есть и более простой путь – посетить магазин готовых фотографий, где можно приобрести желаемое без отчислений от продаж. Это означает, что вы платите за изображение только один раз, а потом можете использовать его в любых целях при соблюдении условий лицензии. Лицензия, как правило, позволяет делать все, что угодно, кроме перепродажи, так что раз купленное изображение можно вставлять в разные проекты. Правда, в магазин может зайти кто угодно, так что есть риск, что «профессиональная деловая женщина» из вашего последнего шедевра на другом сайте будет представлена как «физиотерапевт». Такое бывает.

Если бюджет не позволяет обратиться в магазин изображений, то зайдите на сайты, где исходный материал можно получить бесплатно. Давая фотографам и иллюстраторам-любителям возможность делиться друг с другом своими творениями, сообщества типа iStockPhoto (www.istockphoto.com) (рис. 9) и stock.xchng (www.sxc.hu) создали впечатляющие каталоги фотографий и рисунков со средним и высоким разрешением, которые широко распространяются в Web. Некоторые изображения продаются за небольшую цену, другие вовсе бесплатны, но так или иначе полезность таких сайтов неоспорима.

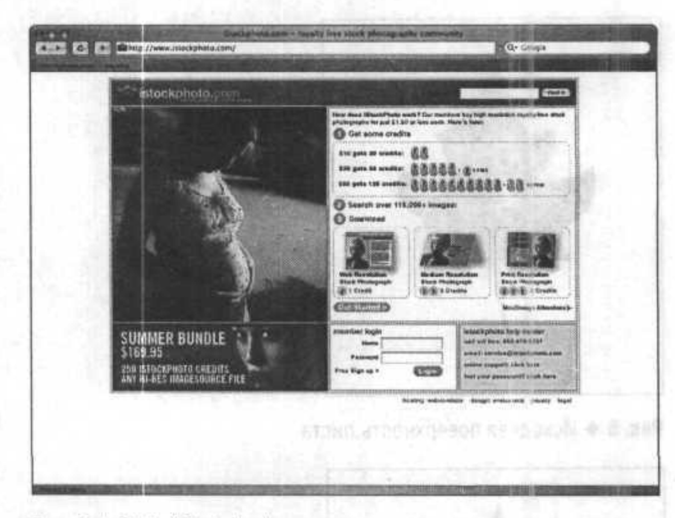


Рис. 9 ✦ Сайт iStockphoto приглашает художников поделиться своими работами



Рис. 10 ✦ На сайтах типа Creative Commons предлагаются произведения искусства, являющиеся всеобщим достоянием

Поиск альтернативных источников

Если вы готовы раскинуть сети пошире, то можно найти и другие источники изображений без отчислений от продаж. В некоторых странах закон об авторском праве распространяется на несколько прошлых десятилетий, но обычно момент, когда произведение искусства становится общественным достоянием, отстоит от момента его создания не слишком далеко. Так, в США в середине 2004 года авторским правом уже не защищались произведения, опубликованные ранее 1923 года. Открыты для использования были также работы, опубликованные в период между 1923 и 1963 годом, если авторское право не было специально продлено.

Работы, являющиеся всеобщим достоянием, можно использовать в любых целях, коммерческих или личных, так что если вам нужно изображение, созданное давно, то стоит подумать и о таком методе. Чтобы найти подходящее произведение, достаточно набрать в поисковой системе запрос public domain. В библиотеках и художественных галереях есть каталоги общедоступных творческих работ. В вашей местной библиотеке может быть даже составленный там же каталог общедоступных фотографий.

Но вовсе необязательно залезать так далеко в прошлое, поскольку есть проекты, ставящие себе целью сделать всеобщим достоянием и новые работы. Так, организация Creative Commons (http://creativecommons.org) «стремится расширить круг произведений искусства, доступных всем для совместного использования и в качестве основы для новых работ» (рис. 10). Хотя в 2004 году основной задачей сайта было убедить художников делиться своими произведениями, вы найдете там работы, опубликованные на условиях лицензии Creative Commons. Впрочем, пользуйтесь ими осторожно, поскольку лицензия предусматривает различные уровни, и не все можно просто так стянуть.

Замечания об авторских правах

Если работа не выставлена на продажу, но вам она абсолютно необходима, то у автора (а чаще у издателя) всегда можно испросить разрешение на использование. В худшем случае он ответит отказом. Позаботьтесь о том, чтобы правообладатель дал разрешение в письменной форме, только после этого вы сможете пользоваться работой для оговоренных целей.

Отметим, что законодательство об интеллектуальной *собственности* запутано и часто *меняется*. Новые творческие работы защищены законом об авторских правах с момента создания вне зависимости от того, поместил ли автор знак копирайта.

Изучите законы, действующие в вашей стране. Если вы не полностью уверены, что можете воспользоваться той или иной работой, получите разрешение или примите иные меры. Воспользовавшись без разрешения работой, защищенной авторским правом, вы рискуете попасть под суд.

СОВЕТ

Система поиска по сайту Creative Commons доступна по адресу http://creativecommons.org/getcontent.

СОВЕТ

Вот несколько полезных ссылок на правительственные ресурсы, посвященные законам об авторских правах: сайт агентства по авторским правам США (www.copyright.gov), сайт с текстом канадского закона об авторских правах (*http://laws.justice.gc.ca/en/C-42*), страница патентного бюро Великобритании, посвященная закону об авторских правах (www.patent.gov.uk/copy).

Вердикт: нарисованный облик

«Береговой бриз» – впечатляющий пример того, что можно сделать, имея капельку таланта и много свободного времени. По оценке Ши, на создание графики для этого дизайна было потрачено более 80 часов работы в Photoshop.

О клиенте, который согласился бы платить за такое времяпрепровождение, можно только мечтать. Для коммерческого проекта аналогичного результата можно было бы достичь с помощью фотографии. Пользуясь подходящими инструментами, удалось бы создать идентичную атмосферу, не подрывая бюджет.

Нарисованные вручную картинки, конечно, отличаются, но на практике именно бюджет определяет, где надо остановиться в работе над проектом. Придавать неповторимость сайту лучше за счет деталей и мелких штрихов.

Дизайнер **Майкл Пик** (Michael Pick)

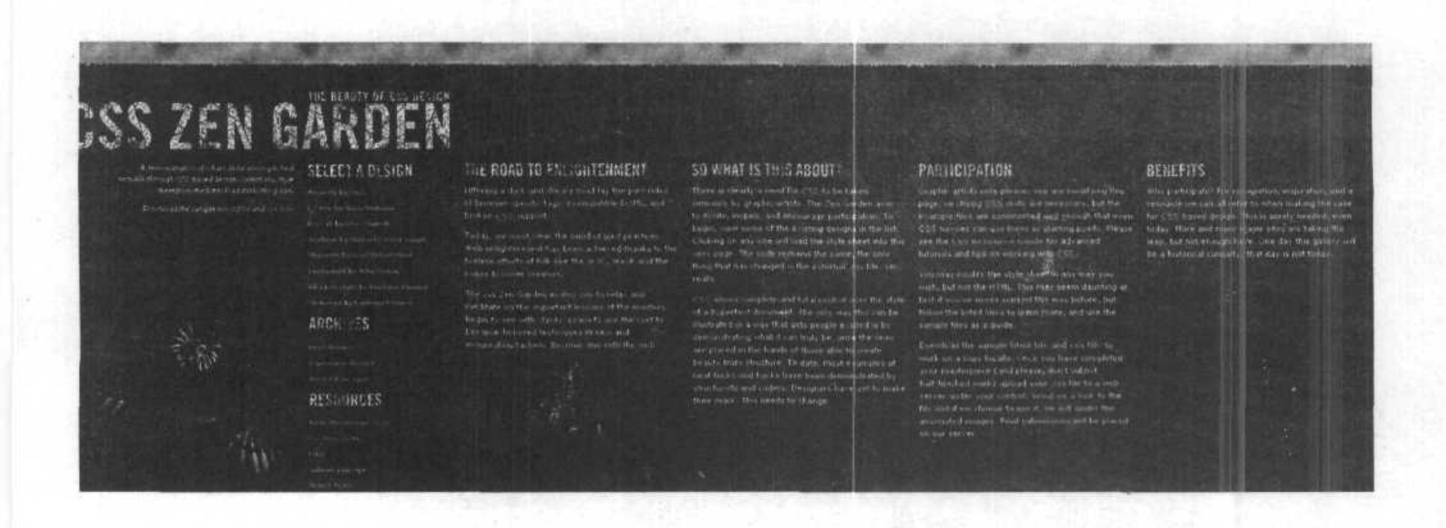
www.csszengarden.com/019

Что там в глубине

Трехмерное пространство на двумерной плоскости

Игра ума и ассоциативные образы – именно так у Майкла Пика рождаются идеи. В данном случае он начал с размышлений об общественных парках в Нью-Йорке и отснял несколько фотографий. Обрабатывая их в Adobe Photoshop, он вдруг заметил, что получающийся результат напоминает муравейник. Это навело его на мысль изобразить не традиционный наземный парк, а некую подземную сцену.

Приняв такое решение, он стал думать о том, что еще можно найти под землей. Нахлынувший поток образов вдохновил его на создание страшноватого, странного, но не лишенного своеобразной красоты дизайна «Что там в глубине».



ПРИМЕЧАНИЕ

Технология VRML никуда не делась. Она все еще развивается и используется в некоторых ситуациях. Найти информацию о версии VRML с открытыми исходными текстами, OpenVRML, можно на сайте www.openvrml.org.

ПРИМЕЧАНИЕ

Подробнее о QuickTime VR можно прочитать на странице www.apple.com/quicktime/qtvr.

ПРИМЕЧАНИЕ

Исследованию и продвижению открытых стандартов на трехмерную графику для Web посвящен проект Web3D Consortium (www.web3d.org).

Создание трехмерной сцены

В большинстве веб-страниц упор делается на том, чтобы представить информацию красиво и в удобной для работы форме. Хотя от дизайнеров часто требуют новых и свежих идей, следует признать, что значительная доля сайтов по необходимости организуют графику и текст на двумерной плоскости.

Есть и исключения. На очень ранней стадии развития Всемирной паутины появилась технология VRML (Virtual Reality Modeling Language – язык моделирования виртуальной реальности). Нанося отдельные точки в моделируемом окружении, разработчик мог создать трехмерную модель для размещения на веб-странице. Для просмотра таких моделей посетителю был необходим подключаемый VRML-модуль.

Чуть позже компания Apple разработала программу QuickTime VR, которая позволяла «сшивать» последовательность панорамных фотографий. Для просмотра результата на веб-странице тоже был необходим подключаемый модуль.

Конечно, трехмерные анимации и игры можно просматривать на веб-странице, если это поддерживается самим браузером или подключаемым к нему модулем, как, например, Flash, ShockWave и модули для других аудио-, видео- и смежных форматов.

Но лишь изредка нам выпадает удача исследовать на глубину «родную» для Web среду – безо всяких инструментов для поддержки виртуальной реальности и прочих специальных программ. Не ограничивая свое воображение двумерной областью, мы можем, как Пик, копнуть поглубже и создать более реалистичную картину.

Моделирование трехмерной среды – это обширная область, в которой ведутся сложные научные исследования. Неплохой отправной точкой для изучения этой темы может стать разработка компьютерных игр. Понятно, что по мере того, как игры становятся все более реалистичными и зрительно убедительными, знакомство с принципами их проектирования может многому научить дизайнера.

Как устроены детали

Выдающаяся графика работы «Что там в глубине» так сильно поражает зрителя благодаря проработке деталей. В темных глубинах мы видим объекты, которым положено находиться под землей, хотя мы не сразу понимаем, что именно подземелье нам и показывают.

Одна из обязанностей дизайнера – увидеть мир настолько детально, чтобы адекватно воспроизвести его в своей работе или, как в случае «Что там в глубине», создать *ощущение* окружающей обстановки.

Значит, вы должны научиться понимать детали, а не только видеть их. Входя в комнату, вы сразу же замечаете очевидное: стены, окна (или их отсутствие) и мебель.

Но как насчет текстуры? Трещины в дереве, отслоившаяся краска, выкрошившаяся штукатурка, треснутое или выцветшее стекло? А много ли внимания вы обращаете на свет и тень? Что можно сказать об общем впечатлении от комнаты? Она мрачная? Светлая? Просторная? Тесная? Отвечая на эти вопросы, вы раскроете не сразу бросающиеся в глаза детали, благодаря которым и возникает ощущение реальности.

В дизайне «Что там в глубине» эти детали видны повсюду. В частности, это муравьи, которые и вдохновили автора первоначально (рис. 1).

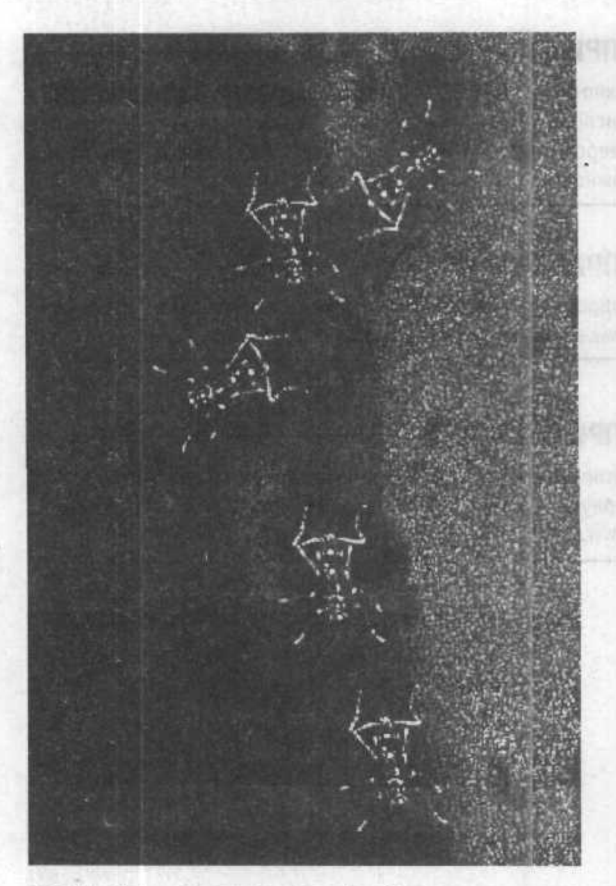


Рис. 1 ✦ Марширующие муравьи (изображение слегка осветлено)

Создание ощущения реальности

Интересно отметить, что для создания реалистичности вовсе не нужна фотографическая точность. Образы, использованные в работе «Что там в глубине», не назовешь фотореалистичными, тем не менее ощущение реальности изображенного очень сильно.

Нужно, чтобы сама графика производила впечатление реальности изображенного. Пытаясь создать изображения, которые не просто копируют природу, а пробуждают мысленные ассоциации, имейте в виду следующее:

✦ обращайте повышенное внимание на края. Эта тема трудно дается начинающим художникам на занятиях по рисованию человеческих фигур. Сначала пытаются просто нарисовать линию, к примеру, женской руки. Но очень быстро выясняется, что не все так просто. Обработка краев подчеркивает объемность очертаний, игру света и тени, различия между позитивным и негативным пространством. Именно это и воспринимается нами как линия (рис. 2);

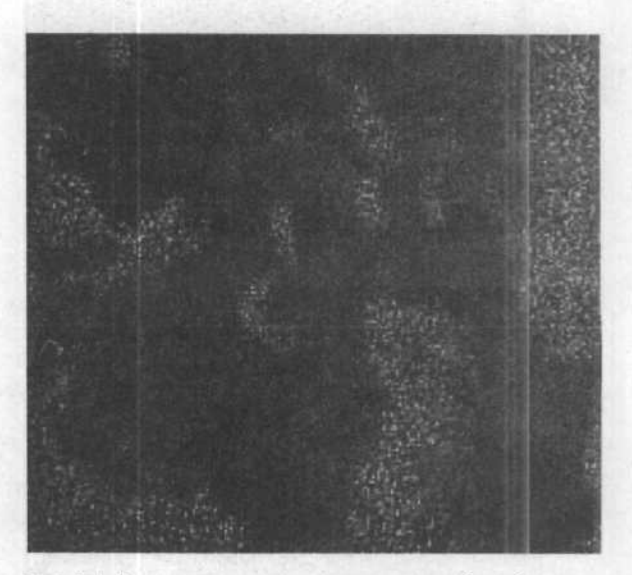


Рис. 2 ✦ Края – это не просто четкие линии. Они образуются в результате игры света и тени, линии и пространства (изображение слегка осветлено)

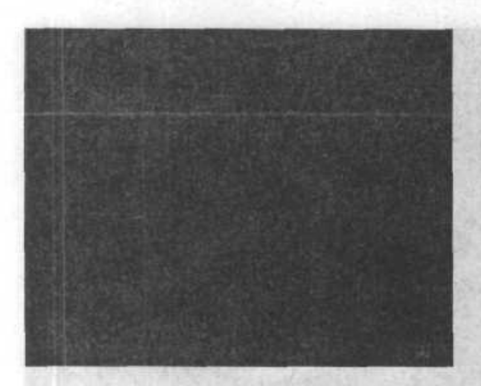


Рис. 3 ✦ Текстура будки управления светофором создает ощущение земли (изображение слегка осветлено)

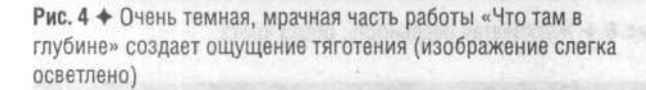


Рис. 4 ✦ Очень темная, мрачная часть работы «Что там в глубине» создает ощущение тяготения (изображение слегка осветлено)

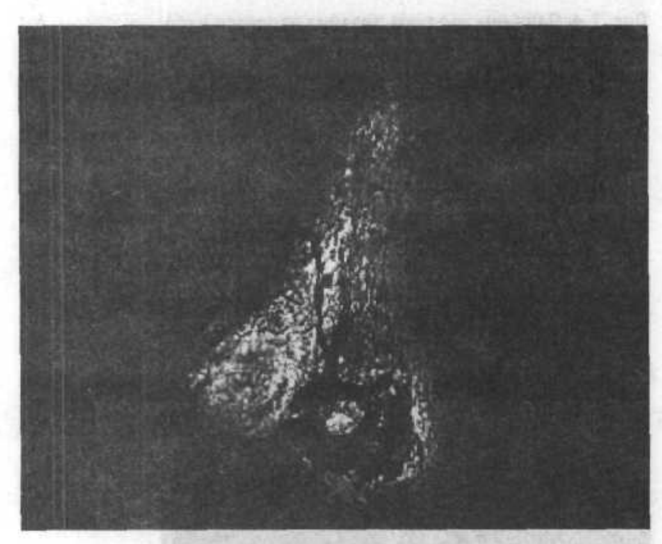


Рис. 5 ✦ Изображение человеческой берцовой кости под землей. Кажется, что кость обладает массой (изображение слегка осветлено)

ПРИМЕЧАНИЕ

Хотя определения массы, твердости и объема в математике и дизайне в чем-то перекрываются, но между ними есть и различия, связанные с тем, как эти понятия применяются в разных отраслях знания.

- не забывайте о форме. Оглядитесь вокруг и найдите предмет, любой предмет. Тщательно осмотрите и определите его форму. В большинстве объектов вы обнаружите множество разных форм, это очень важно. Нельзя сказать, что чашка – это всего лишь окружность. Конечно, в ней есть и окружности, и дуги. Но могут быть и другие формы, которые отличают ее от других чашек;
- текстура изображения создает реалистичное представление о качестве предметов. Чтобы нарисовать почву в работе «Что там в глубине», Пик сделал фотографию будки управления светофором на Хьюстонстрит в Нью-Йорке. Интересно, что текстура будки – совершенно, казалось бы, не относящегося к делу предмета, – наводит на мысли о почве (рис. 3);
- глубина – вот что дает настоящее ощущение объемности, даже в двумерном окружении. В геометрии ось x определяет горизонтальное, а ось y – вертикальное направление. В третьем измерении проходит ось z. Именно с осью z и связана глубина. Восприятие глубины в двумерном дизайне определяется светом и тенью краев. На рис. 2 видно, что более темные области выглядят как углубления на поверхности земли;
- важна и имитация тяжести. В двумерном дизайне этот эффект достигается за счет зрительной тяжеловесности образов. К моделируемому окружению применимы те же принципы, но необходимо еще какое-то визуальное указание на то, что предмет склоняется под своей тяжестью к низшей точки изображения. Самая темная, окрашенная сплошным цветом область в дизайне «Что там в глубине» находится в нижней части подземной сцены, это усугубляет ощущение тяготения (рис. 4).

При создании представления трехмерных объектов на двумерной странице комбинация описанных выше средств создает иллюзию *массы*.

В контексте визуального дизайна массу можно описать как характерный признак трехмерного объекта, призванный создать впечатление, будто он выделяется на фоне окружающего его пространства. Примите во внимание следующие принципы:

- иллюзия твердости создается за счет цвета и текстуры;
- для создания иллюзии объемности важны длина, ширина и глубина.

Масса – это результат сочетания подобных визуальных элементов, в результате чего объект приобретает реалистичные черты (рис. 5).

И, наконец, если взять какой-то природный объект и поместить его в смоделированную среду, то эффект реалистичности может усилиться. В данном случае образ крота вселяет в зрителя жутковатое чувство достоверности остальной части дизайна (рис. 6).

Работа с уровнями

Еще одна сторона, с которой веб-дизайнеры редко сталкиваются, – это работа с уровнями. Имеются в виду отдельные слои в пространстве. Концепция уровней становится понятной, если вспомнить об офисном здании. Каждый этаж – это уровень.

При программировании игр огромное внимание уделяется проработке архитектурных уровней в пространстве игры. Дизайнеры игр часто специально изучают, как архитекторы и дизайнеры окружающей среды подходят к проектированию зданий.

Из архитектуры, дизайна окружающей среды и у разработчиков игр мы можем позаимствовать следующие идеи, важные при моделировании многоуровневого пространства:

- ✦ каждый уровень уникален с точки зрения линий, пространственного расположения и масштаба;
- ✦ каждый уровень, тем не менее, интегрирован в общую картину и тем или иным способом связан с уровнями выше и ниже его;
- ✦ между поверхностями имеются естественные переходы;
- ✦ в дизайне всегда есть фокусная точка;
- ✦ как уже отмечалось, важно передать идею тяготения. Человек *ожидает*, что на предметы действует сила тяжести.

В работе «Что там в глубине» есть три четко выраженных уровня: небо (рис. 7), трава (рис. 8) и подземная область (рис. 9).

Изучая эти изображения, можно понять, как идеи воплощаются на практике. У каждого уровня есть свои особенности. Каждый уровень органично связан с уровнем выше (рис. 10) и уровнем ниже (рис. 11). Фокусной точкой в данном случае служит подземелье.

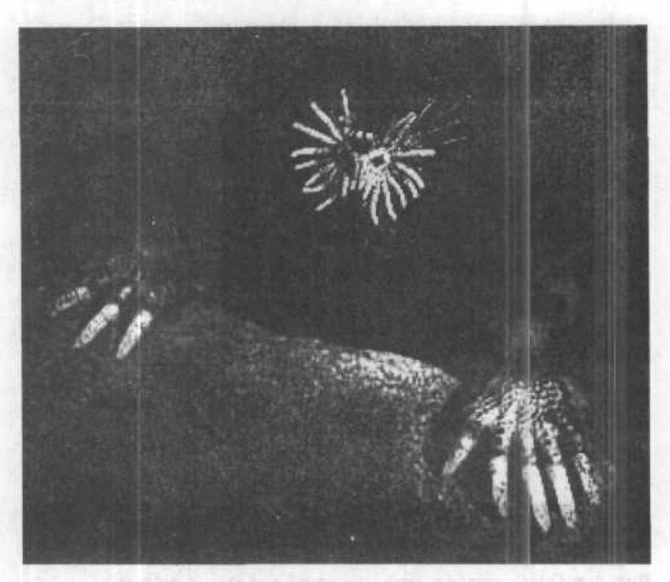


Рис. 6 ✦ Жутковатая реальность: образ крота

Рис. 7 ✦ Верхним уровнем дизайна является небо

Рис. 8 ✦ Средний уровень – трава (изображение слегка осветлено)

ПРИМЕЧАНИЕ

Цветной вариант рис. 7 и 8 см. на вкладке.

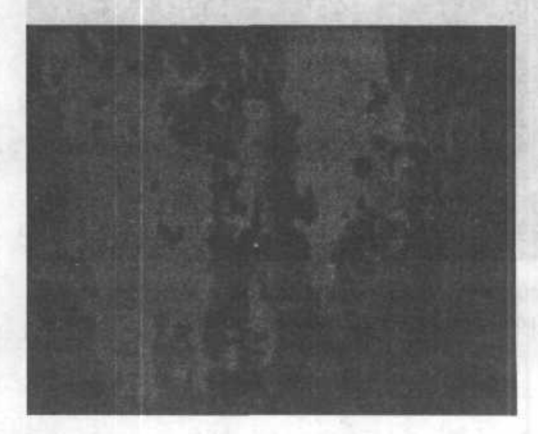


Рис. 9 ✦ Нижняя часть страницы занята подземным уровнем

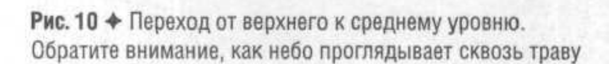


Рис. 10 ✦ Переход от верхнего к среднему уровню. Обратите внимание, как небо проглядывает сквозь траву

Рис. 11 ✦ Переход от травы к подземному уровню достигается за счет постепенного нарастания темных цветов и изображения корней травы, прорастающих из земли (изображение слегка осветлено)

ПРИМЕЧАНИЕ

Цветной вариант рис. 10 и 11 см. на вкладке.



Рис. 12 ✦ Порча шрифта способствует его интеграции в пространство дизайна



Рис. 13 ✦ Использование того же цвета, который применяется в смоделированной сцене, помогает включить нереалистичные объекты в контекст

Интересно, что подземный уровень создает ощущение тяготения за счет своей большей высоты, массивности и более темного цвета – по сравнению с верхним и средним уровнями.

Нереалистичные образы в смоделированном пространстве

Все мы знаем, что ни заголовок, ни текст, ни ссылки под землей не существуют. Однако Пик применил такую технику, при которой абстрактные объекты кажутся естественно интегрированными в дизайн.

Обратите внимание на текст заголовка (рис. 12). Видно, что дизайнер взял некоторый шрифт и сознательно подпортил его. Поцарапанные и испачканные буквы лучше сочетаются с прочей графикой на странице.

Пик также неслучайно выбрал цветовую гамму. Цвет ссылок совпадает с более яркими частями фигуры крота. Это создает ощущение освещенности и более тесно увязывает ссылки с другими частями страницы (рис. 13).

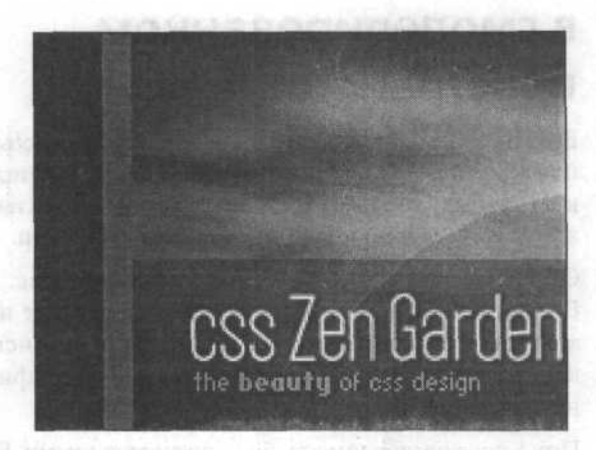
Неслучайно и то, что к ссылкам не применяются никакие цветовые эффекты. При наведении курсора мыши на ссылку она лишь подчеркивается. Это препятствует появлению лишних ассоциаций.

Если бы Пик решил взять для ссылок другие цвета, скажем, голубой цвет неба или зеленый цвет травы, то разрушил бы иллюзию, поскольку придал бы подземному уровню несвойственные ему качества.

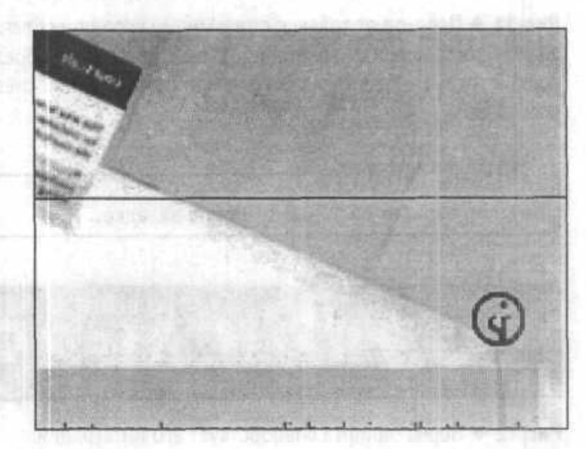
А так целостность дизайна не нарушается.

Уроки трехмерности

Вообще говоря, именно экспериментальная природа сайта CSS Zen Garden позволила исследовать возможности моделирования подземного пространства. Только обладая изобретательным умом и вниманием к деталям Майкла Пика, можно было создать такой уникальный дизайн.



Два океана, с. 174



si6, с. 180



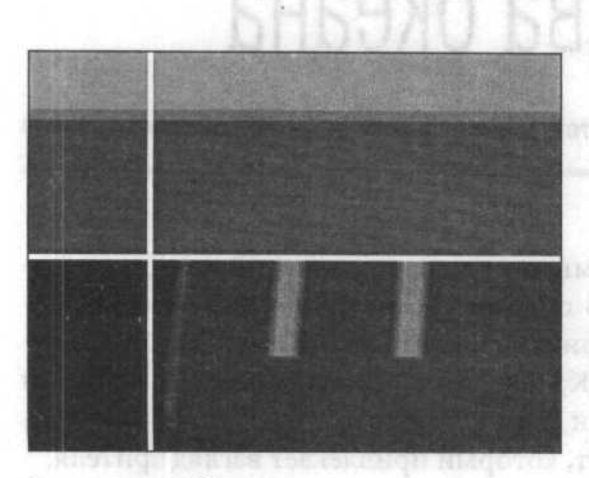
Живым или мертвым, с. 194



Жажда крови, с. 200

5

Типография



Версия один, с. 186



Золотая середина, с. 208

Естественное упрощение форм общения во многом обязано печатному слову. Именно таким способом чаще всего происходит обмен идеями, описываются события, чувства и другие человеческие проявления.

Любой практикующий веб-дизайнер знает, что типография в Web сопряжена с серьезными трудностями. Но это не должно останавливать вас на пути совершенствования своей техники и исследования технологий, с помощью которых можно удовлетворить страсть к форме, стилю, толщине и гарнитуре шрифтов. Эта глава будет посвящена приемам, связанным с типографскими особенностями дизайна.

Дизайнер **Райан Симс** (Ryan Sims)

www.csszengarden.com/085

Два океана

Простота типографии ведет к эффектному дизайну

Изумительная цветовая палитра «Двух океанов» обязана своим происхождением фотографии, которую Райан Симс сделал в Южной Калифорнии. Спокойные, как сам Тихий океан, цвета и образы создают легкий контраст, который привлекает взгляд зрителя.

Работа «Два океана» не кажется кричащей или резкой. Частично это ощущение утонченности создается за счет простоты дизайна. Симс не стал злоупотреблять изображениями, цветами и шрифтами, а последовал философии Хемингуэя: «Уберите все вычурное». Он с удовольствием наблюдал, как дизайн оживает по мере того, как лишнее удаляется, а не добавляется.

Соглашения о шрифтах

Типография – это особая область дизайна. Многие дизайнеры всю жизнь занимаются исследованием, применением и созданием шрифтов. И хотя единого подхода к разработке эффектного шрифта не существует, но есть важные соглашения, о которых должен знать каждый дизайнер.

Форма шрифта

Форма шрифта – это общий термин, включающий такие характеристики шрифта, как насыщенность, кегль и наклон.

Насыщенность шрифта – это мера толщины отдельных литер. В CSS этой характеристике соответствует свойство `font-weight`. В табл. 1 описан смысл различных значений этого свойства.



Рис. 1 ✦ Когда промежуточных насыщенностей нет, браузер пытается сделать все, что в его силах, но часто вынужден ограничиваться только нормальным или полужирным начертанием

Разумеется, тут полно нюансов, и самый важный из них состоит в том, что не во всех шрифтах есть промежуточные варианты, согласующиеся с требованиями CSS (рис. 1). Более того, даже если шрифт обладает вариантом с нужной насыщенностью, браузер не всегда сможет его отобразить.

На практике можно всегда быть уверенным в наличии по крайней мере двух видов начертания шрифта (`normal` и `bold`). На рис. 1 показано, как браузер, в данном случае Mozilla, выводит текст предположительно шрифтами разной насыщенности. В реальности оказывается, что насыщенностей всего две: полужирная и нормальная.

century condensed
century normal
century expanded

Рис. 2 ✦ Узкий, нормальный и широкий варианты начертания шрифта

Под *кеглем шрифта* понимается ширина литер, составляющих этот шрифт. Встречается узкое (condensed), нормальное и широкое (expanded) начертание (рис. 2). В CSS для этой характеристики есть свойство `font-stretch`. Однако оно не используется, потому что ни один из распространенных браузеров его не поддерживает.

Таблица 1 ✦ Насыщенности шрифтов в CSS

Свойство и значение	Назначение
font-weight: normal	Отображает шрифт в нормальном начертании. Соответствует числовому значению насыщенности 400
font-weight: bold	Отображает шрифт в полужирном начертании. Обычно соответствует насыщенности 700, но для некоторых шрифтов полужирное начертание может начинаться с 600
font-weight: (300)	Отображает шрифт более тонкий, чем при нормальном начертании, при шкале от самого тонкого (100) до самого жирного (900)
font-weight: bolder	Отображает шрифт в начертании, несколько более насыщенном, чем унаследованное. Например, если шрифт в абзаце уже полужирный, а в стиле для элемента strong внутри этого абзаца указано bolder, то такой элемент будет казаться еще жирнее. Единственное исключение – когда насыщенность родительского элемента и так уже равна 900; в этом случае bolder не действует, насыщенность останется без изменения
font-weight: lighter	Отображает шрифт в начертании, несколько более светлом, чем унаследованное. Если унаследованная насыщенность шрифта составляет 300, то вычисленное значение lighter будет равно 200, если только существует вариант шрифта с такой насыщенностью. Если насыщенность родительского элемента равна 100, то значение lighter не действует, насыщенность останется без изменения
font-weight: inherit	Насыщенность шрифта наследуется от родительского элемента

Еще одной характеристикой шрифта является *наклон*, он соответствует свойству CSS font-style. Существует два вида наклона:

- ✦ Italic (курсив) – специально разработанная модификация прямого шрифта с дополнительными «завитушками», как при рукописном начертании;
- ✦ Oblique (наклонный) – модификация прямого шрифта, отличающаяся от курсива, поскольку она разработана не специально, а просто повторяет рисунок прямого шрифта, но с наклоном основных штрихов. Многие люди употребляют термин *italic* там, где на самом деле имеется в виду *oblique*.

У свойства font-style есть четыре возможных значения: normal, italic, oblique и inherit.

ПРИМЕЧАНИЕ

Если на вашем компьютере установлен полужирный, курсивный или наклонный вариант некоторого шрифта, то браузер обычно им и воспользуется. Если же такого варианта нет, то браузер сымитирует его, программно модифицировав обычный шрифт. Разница часто заметна острому глазу дизайнера.

Контрастность шрифта

Контрастностью шрифта называется соотношение между тонкими и толстыми линиями одной литеры. У разных шрифтов эти пропорции меняются весьма существенно, но есть и такие шрифты, где все линии имеют постоянную толщину.

В стандартном полиграфическом дизайне принято использовать один шрифт для заголовков, а другой – для основного текста. Как правило, в заголовках применяется шрифт без засечек, поскольку округлые, обычно слабоконтрастные литеры привлекают больше

ПРИМЕЧАНИЕ

У многих складывается впечатление, что в Web дело обстоит прямо противоположным образом: на экране лучше выглядят шрифты без засечек. Но споры продолжаются. Подробнее об этой полемике можно прочитать на странице www.digital-web.com/news/2004/06/serif_vs_sanserif.

внимания. Шрифты с засечками долгое время считались предпочтительными для основного текста, особенно если он состоял из длинных фрагментов. Большинство книг напечатано именно такими шрифтами, но вопрос в том, следует ли применять ту же практику в Web, все еще дебатируется.

Цвет шрифта

Под *цветом шрифта* (type color) в полиграфии понимают общий оттенок блока текста. У светлых шрифтов типа Helvetica Light светлый цвет, поскольку наносится очень мало краски; насыщенные же шрифты типа Cooper или Arial Black выглядят гораздо темнее, потому что краски наносится больше. Но цвет шрифта зависит не только от гарнитуры, на восприятие влияют и другие факторы, например межбуквенный интервал (кернинг) и межстрочный интервал (интерлиньяж).

Помимо контрастности, для зрительного восприятия шрифта важен еще цвет, которым печатаются литеры. *Оттенок шрифта* (type hue) играет заметную роль, так как наряду с контрастными шрифтами дизайнер может еще выбирать и контрастные цвета.

Вообще говоря, не рекомендуется злоупотреблять многоцветием при выборе шрифтов. Бывают, конечно, случаи, когда наличие большого числа цветов оправдано, – скажем, на сайтах, предназначенных для детей. Но обычно переизбыток цветов свидетельствует о непрофессионализме.

Шрифты в «Двух океанах»

Стремление Симса к простоте отчетливо видно на примере работы «Два океана». Весь HTML-текст отображается шрифтом Trebuchet, а на случай его отсутствия указано еще несколько шрифтов без засечек:

```
body {font: 11px; «trebuchet ms», arial,
  helvetica, sans-serif;}
```

Trebuchet – это шрифт без засечек, входящий в базовый комплект шрифтов Miscrosoft для Web и поставляемый также в комплекте с Microsoft Office. Это значит, что он распространен достаточно широко, хотя предусмотреть его отсутствие тоже необходимо.

Выбор оттенка и контрастности

Выбор оттенка и контрастности шрифта критически важен – не только для поддержания целостности дизайна, но и с точки зрения *удобочитаемости*, то есть легкости восприятия текста.

По традиции рекомендуется выбирать высококонтрастные цвета для фона и переднего плана, например черный текст на белом фоне или наоборот. Предполагается, что наряду с другими приемами это повысит удобочитаемость.

В «Двух океанах» Симс отошел от этой традиции и выбрал для основного текста очень изысканный желто-коричневый цвет:

```
color: #9F927F;
```

Контраст с белым фоном в этом случае относительно слабый (рис. 3). Интересно, что ссылки и в основном тексте, и по краям отображаются более темным оттенком синего цвета. Многие дизайнеры предпочли бы использовать такой цвет для основного текста из-за более высокой контрастности (рис. 4).

Надо признать, что при печати основного текста синим цветом контрастность улучшилась, но утонченный контраст, достигаемый за счет использования желто-коричневого цвета, выглядит приятнее. Именно выбор этого цвета и придает странице такой успокоительный вид, из-за чего и читать ее проще.

Более того, употребление синего цвета для выделения ссылок как внутри текста, так и в списке добавляет всему дизайну контрастности. И заодно делает страницу более интересной – подход тонкий, но в данном случае весьма эффективный.

Контрастность играет роль и при выборе инверсных цветов для заголовка. Текст внутри изображения отображается белым шрифтом на полупрозрачном фоне поверх фотографии (рис. 5). А сразу под ним расположен подзаголовок, напечатанный тоже белым шрифтом, но на темном желто-коричневом фоне. Это создает интересный переход форм и движение, усиливающее общий эффект (рис. 6).

Текст в основной части страницы напечатан шрифтом размером 11 пикселей, равно как и ссылки в главном списке. Для дополнительных же ссылок выбран шрифт размером 9 пикселей, чтобы отразить их меньшую значимость. Удивительно, что может сделать разница всего в два пикселя, когда используется в сочетании с переменной контрастностью (рис. 7).

Рис. 3 ✦ В этом дизайне используется слабоконтрастный желто-коричневый шрифт

Рис. 4 ✦ Улучшение контрастности за счет выбора более темного (синего) оттенка из той же палитры не так хорошо соответствует общей теме дизайна

СОВЕТ

Без сомнения, выбор слабоконтрастных цветов может вызвать трудности у людей с ослабленным зрением. Для профессиональных дизайнерских работ, рассчитанных на широкую аудиторию, лучше придерживаться более традиционных решений в плане цветов и контрастности, чтобы страница была удобочитаемой.

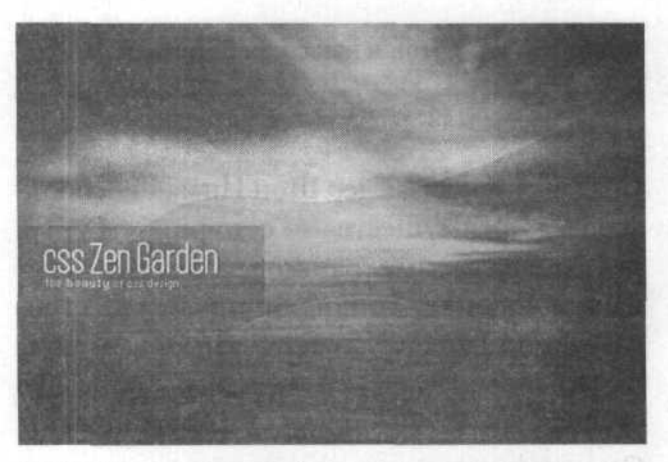


Рис. 5 ✦ Контрастность шрифта в заголовке «Двух океанов»

Рис. 6 ✦ Инверсные цвета в тексте, расположенном ниже заголовка

ПРИМЕЧАНИЕ

Цветной вариант рис. 5 и 6 см. на вкладке.

Рис. 7 ✦ Когда визуальный эффект создается разницей в размере и контрастности шрифта, 2 пикселя кажутся больше, чем на самом деле

Рис. 8 ✦ Графический текст набран заглавными буквами и помещен на другой фон. Он выделяется на странице, поскольку больше нигде такая комбинация цветов не встречается

Использование размера и насыщенности шрифта для создания контрастности

Возвращаясь к заголовку на рис. 5, хочется обратить ваше внимание на то, что в этом графическом тексте Симс для создания контрастности воспользовался размером и весом шрифта. Слово css набрано маленькими буквами и контрастирует с начальными заглавными буквами в словах Zen Garden. Подзаголовок ниже дает прекрасный пример создания контраста за счет насыщенности: слово beauty в нем выделено полужирным шрифтом из окружающего текста, отображаемого обычным шрифтом.

Тот же прием прослеживается и в картинке непосредственно под фотографией, где первые семь слов напечатаны полужирным шрифтом, а остальные – нормальным (рис. 6).

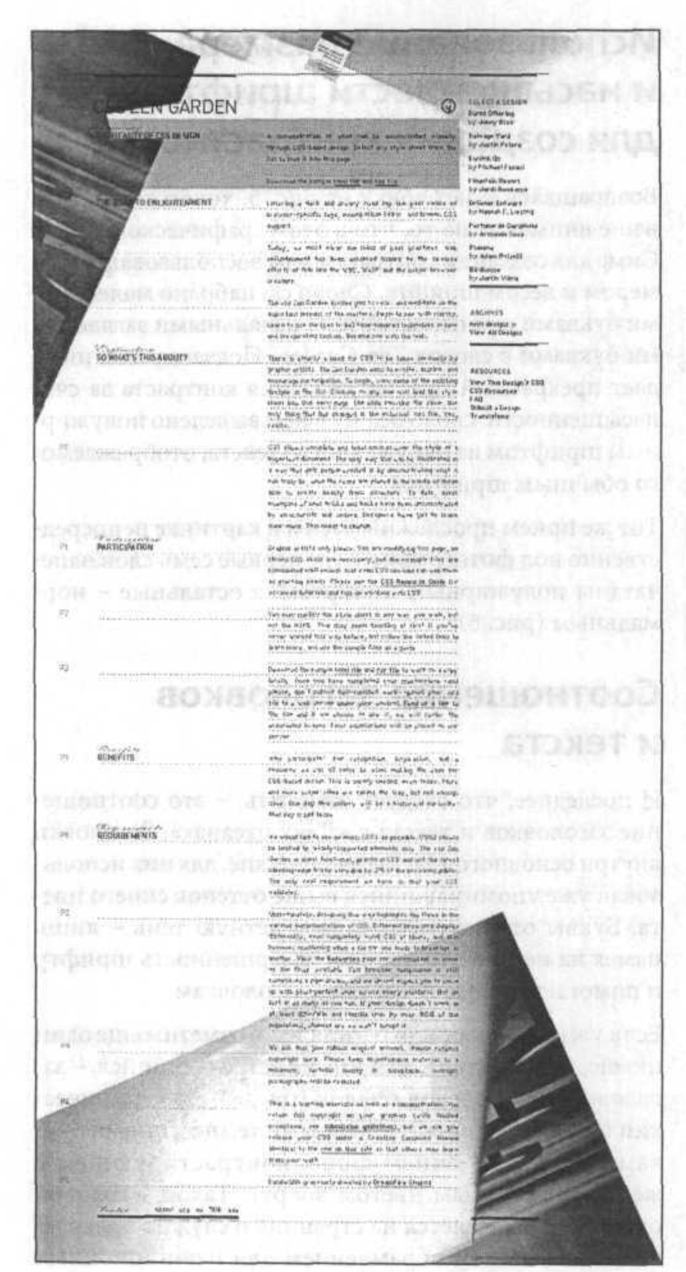
Соотношение заголовков и текста

И последнее, что следует отметить, – это соотношение заголовков и текста в «Двух океанах». Заголовки внутри основного текста графические, для них использован уже упоминавшийся выше оттенок синего цвета. Буквы отбрасывают едва заметную тень – лишь намек на нее, но это придает завершенность шрифту и помогает привлечь взгляд к заголовкам.

Если уж мы заговорили о взгляде, то отметим еще один нюанс, более нигде в дизайне не встречающийся, – заголовки для разделов ссылок (рис. 8). Это графический текст, набранный заглавными темно-синими буквами на светло-синем фоне, контрастирующем с желто-коричневым цветом вокруг. Такой заголовок отчетливо выделяется на странице и служит прекрасным визуальным обрамлением для навигационных ссылок.

Учитесь простоте

Хотя на первый взгляд дизайн «Двух океанов» кажется бесхитростным, но эта обманчивая простота создает спокойное и беззаботное настроение. Тонкие дополнительные штрихи, особенно в использовании шрифтов, – это урок, который не стоит забывать при работе над собственными проектами.



Дизайнер **Шон Инман** (Shaun Inman)
www.csszengarden.com/044

si6

Проблема шрифтов и как с ней бороться

Создавая si6, дизайнер Шон Инман воспользовался преимуществами сеточной структуры. Воображаемые границы разделяют макет на три колонки. Тонкие линии между строками текста подчеркивают взаимосвязь между заголовками в левой колонке и абзацами в средней части, к которым они относятся.

Сам текст аккуратно расположен точно поверх линий, нарисованных на фоновом изображении. Благодаря такой точности страница выглядит безупречно, но это впечатление легко разрушить, если пользователь изменит масштаб текста (а из-за малого кегля шрифта и слабой контрастности это может оказаться необходимым).

СОВЕТ

Microsoft публикует списки шрифтов, поставляемых с различными приложениями (www.microsoft.com/typography/fonts/default.aspx). Но поскольку пользователи могут удалять или просто не устанавливать шрифты, то все равно нет никакой гарантии, что нужный шрифт будет доступен.

СОВЕТ

В 2001 году был проведен опрос на тему, какие шрифты используются *наиболее* часто. Хотя его результаты нельзя считать непогрешимыми, так как доля ошибочных ответов неизвестна, но все же основой для сравнения они служить могут. Подробнее об этом опросе читайте в статье «Code Style Font Sampler» (www.codestyle.org/css/font-family).

Ограниченность набора шрифтов

Быть может, наибольшую досаду при работе со шрифтами в Web вызывает тот факт, что выбор доступных шрифтов неширок и непредсказуем, учитывая, что пользователь может устанавливать на свой компьютер одни шрифты и удалять другие. Маловероятно, что в обозримом будущем в этой области что-то кардинально изменится.

Если не считать шрифтов, используемых во Flash-анимациях или встроенных в графические изображения, то вся типография в Web базируется на тех шрифтах, что есть на компьютере пользователя. Однако пользователи по большей части не являются дизайнерами, поэтому на их машинах редко можно встретить шрифты помимо тех, что поставляются в комплекте с операционной системой или приложениями. К примеру, на тех машинах, где установлен Microsoft Office, выбор шрифтов шире, чем на машинах, где этот пакет не стоит.

Системы Windows и Mac OS X комплектуются разными наборами шрифтов. В какой-то мере они пересекаются, но если еще учесть и различные варианты Unix (включая и Linux), то окажется, что уверенным можно быть лишь в том, что ни в чем нельзя быть уверенным. В табл. 1 перечислены некоторые шрифты, обычно присутствующие на каждой из трех платформ.

Таблица 1 ✦ Наиболее распространенные шрифты для Web по платформам

	Windows	Macintosh	Unix/Linux
С засечками	Book Antiqua, Bookman Old Style, Garamond	New York, Palatino, Times	Bitstream, Vera Serif, New Century Schoolbook, Times, Utopia
Без засечек	Arial Narrow, Century Gothic, Lucida Sans Unicode, Tahoma	Charcoal, Chicago, Geneva, Helvetica, Lucida Grande	Bitstream Vera Sans, Helvetica, Lucida
Моноширинные	Courier, Lucida, Console	Courier, Monaco	Bitstream Vera Mono, Courier

В 1996 компания Microsoft выпустила комплект шрифтов, спроектированных специально для просмотра на экране. Это базовые шрифты для Web, их можно было скачать бесплатно, поэтому они получили широкое распространение. В наши дни они обычно устанавливаются вместе с продуктами Microsoft. Включенные в этот комплект шрифты перечислены в табл. 2.

Таблица 2 ✦ Основные шрифты для Web от Microsoft

С засечками	Без засечек	Моноширинные	Специальные
Georgia Times New Roman	Arial, Arial Black, Trebuchet MS, Verdana	Courier New, Andale Mono	Comic Sans MS, Impact, Webdings

Вообще говоря, считается безопасным предполагать в дизайне, что эти шрифты существуют, но, когда речь идет о Web, уверенным ни в чем быть нельзя. Именно по этой причине в CSS разрешается указывать несколько шрифтов и даже общее семейство на тот случай, если ни одного из них в машине пользователя не окажется.

ПРИМЕЧАНИЕ

К сожалению, в 2002 году бесплатное распространение базового комплекта шрифтов для Web на официальном сайте Microsoft было прекращено, но (благодаря либеральной лицензии) они выложены по адресу http://corefonts.sourceforge.net. Поскольку эта страница предназначена для пользователей Linux, то найти версии для Windows там нелегко, поищите фразу «Original unaltered.exe files».

Общие семейства шрифтов

Инман хотел, чтобы в si6 использовались шрифты с гарнитурой Geneva, применяемые в Mac. Поскольку он знал о том, чем рискует, выбирая шрифт, отсутствующий в Windows, то включил в список предпочтительных шрифтов и альтернативы:

```
body {
  font: 9px/16px Geneva,Arial,Tahoma,sans-serif;
}
```

В данном объявлении первые два значения – это кегль шрифта (9 px) и интерлиньяж (16 px). Затем перечислены допустимые шрифты в порядке предпочтительности (рис. 1). Если браузер не сможет найти шрифт Geneva, то он сначала будет искать Arial, затем Tahoma и в самом крайнем случае, когда все другие возможности исчерпаны, шрифт без засечек, указанный в настройках по умолчанию.



Рис. 1 ✦ Как выглядит si6 при использовании разных шрифтов: Geneva (сверху), Arial и Tahoma

Общие семейства шрифтов встроены в CSS, и совместимые браузеры автоматически находят шрифты, соответствующие каждому семейству (хотя пользователь может отменить выбор браузера). Несмотря на то что от дизайнера не требуется во что бы то ни стало указывать общее семейство в каждом объявлении шрифтов, лучше это сделать (если, конечно, вы не являетесь непоколебимым приверженцем Times New Roman).

Times New Roman

Georgia

Garamond

Рис. 2 ✦ Примеры шрифтов с засечками: Times New Roman, Georgia и Garamond

Helvetica

Futura

Lucida Sans Unicode

Рис. 3 ✦ Примеры шрифтов без засечек: Helvetica, Futura и Lucida Sans Unicode

Courier

Monaco

Andale Mono

Рис. 4 ✦ Примеры моноширинных шрифтов: Courier, Monaco и Andale Mono

Comic Sans

Apple Chancery

Brush Script MT

Рис. 5 ✦ Примеры рукописных шрифтов: Comic Sans, Apple Chancery и Brush Script MT

Impact

Papyrus

Рис. 6 ✦ Примеры фантазийных шрифтов: Impact и Papyrus

Ниже перечислены все общие семейства с кратким описанием каждого.

serif

В шрифтах семейства `serif` есть декоративные засечки, то есть небольшие черточки на концах штрихов (рис. 2). Почти всегда назначаемым по умолчанию шрифтом семейства `serif` является Times New Roman, если только пользователь явно не указал иное.

sans serif

Шрифты семейства `sans-serif` не имеют засечек (рис. 3). Почти всегда назначаемым по умолчанию шрифтом семейства `san-serif` является Arial.

monospace

Шрифты семейства `monospace` имеют постоянную ширину: каждая литера занимает одинаковое пространство (рис. 4). Так, например, ширина буквы i такая же, как буквы m. Такие шрифты бывают полезны, когда надо напечатать листинг программы, в котором символы должны находиться друг под другом. Почти всегда назначаемым по умолчанию шрифтом семейства `monospace` является Courier.

cursive

Шрифты семейства `cursive` имитируют рукописное написание и применяются по большей части в заголовках (рис. 5). По умолчанию назначаются разные шрифты, но чаще всего это Comic Sans.

fantasy

Фантазийные шрифты применяются исключительно в декоративных целях – как правило, в заголовках (рис. 6). Поскольку к этой категории можно отнести очень много разных шрифтов, то предсказать, как будет выглядеть шрифт заданного размера в конкретном браузере, невозможно. Поэтому для серьезного дизайна ими не пользуются.

Красиво, но...

Если, занимаясь дизайном сайтов, вы в трех из четырех случаев пользуетесь шрифтами Verdana и Georgia, то иногда выбор шрифта, который заведомо отсутствует на машинах большинства посетителей, может показаться забавным способом привнести некоторое оживление в свою работу. Нужно только указать

резервные шрифты для большинства пользователей, у которых нет основного, и тогда вы сможете наслаждаться новыми возможностями, позабыв об ограничениях.

Следующие комбинации хорошо работают, поскольку указанные шрифты более-менее похожи; первый шрифт в списке выглядит ново и неожиданно, а следующие – безопасная подстраховка:

```
body {
  font-family: "Lucida Grande", "Lucida Sans
Unicode", Verdana,
  sans-serif;
  }
h2 {
  font-family: Garamond, "Times New Roman",
 serif;
 }
dt, li {
  font-family: Optima, Arial, sans-serif;
 }
 .sidebarCallout {
  font-family: "Century Gothic", Verdana, sans-
serif;
 }
```

Выше вы уже видели, что Инман в качестве основного шрифта указал Geneva, а в качестве резервных – Arial и Tahoma. Такая комбинация особенно удачна, поскольку у всех трех шрифтов примерно одинаковая x-высота. *X-высота* определена в книге Rob Carter, Ben Day и Philip Meggs «Typographic Design: Form and Communication» 3-е издание (John Wiley & Sons, 2002) как «высота строчных букв, исключая верхние и нижние выносные элементы. Чаще всего измеряется по строчной букве x».

Возможно, вы обратили внимание на то, что некоторые шрифты кажутся крупнее других, хотя указан один и тот же кегль; например, с кеглем 16 px Verdana кажется заметно больше других шрифтов того же размера (рис. 7). Составляя список шрифтов, следует иметь в

Impact
Chicago
Verdana
Trebuchet MS
Lucida Grande
Geneva
Eurostile
Andale Mono
Helvetica
Arial
Arial Narrow
American Typewriter
New York
Monaco
Big Caslon
Palatino
Optima
Times New Roman
Georgia
Futura
Apple Chancery
Bell Gothic
Myriad Pro
COPPERPLATE
Gill Sans
Courier New
Baskerville
Brush Script MT

Рис. 7 ✦ Разные шрифты с кеглем 16 px и различной x-высотой. Обратите внимание, как отличаются шрифты в верхней и нижней части списка

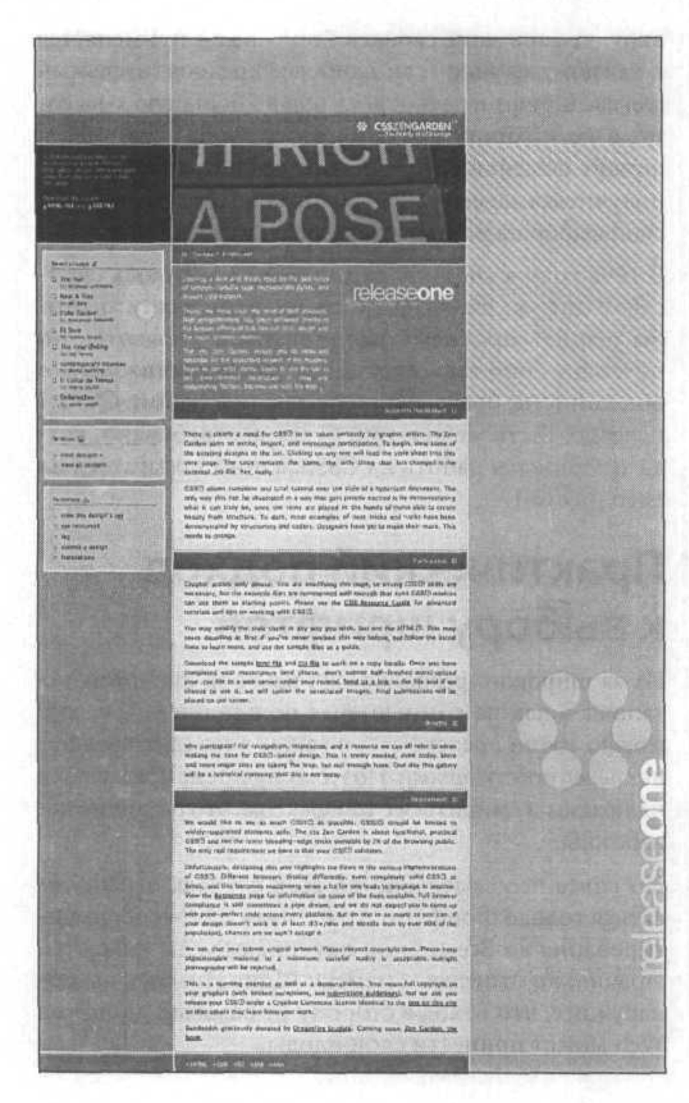
виду, что значение свойств font-size применяется к каждому из них; если наиболее предпочтительный для вас шрифт имеет относительно большую x-высоту, а на компьютерах пользователей установлен шрифт поменьше, то в некоторых ситуациях может получиться нечитаемый текст. Поэтому проверяйте свой выбор на разных браузерах и платформах.

В 1998 году в спецификации CSS2 появилось свойство font-size-adjust специально для того, чтобы решить проблему неодинаковой x-высоты, но поскольку оно так никогда и не было реализовано в большинстве браузеров, то из спецификации CSS 2.1 исчезло. Быть может, оно будет востребовано, когда завершится работа на CSS3, а пока полагаться на него не стоит.

Практический подход к выбору шрифтов

Из-за широкого разнообразия наборов шрифтов, установленных на компьютерах пользователей, невозможно точно предсказать, в каком виде сайт предстанет перед посетителями. Поэтому лучше не предаваться иллюзиям и смириться с тем фактом, что различия неизбежны.

Но вовсе необязательно во всех случаях ограничиваться только шрифтом Arial – пути для экспериментирования во Всемирной паутине все еще открыты. Правильно отнесясь к намеку Шона Инмана, вы обнаружите, что отход в сторону от базовых шрифтов Web может принести свои плоды.



Дизайнер **Дидье Хильхорст** (Didier Hilhorst)

www.csszengarden.com/035

Версия один

Как примириться с безумием размеров шрифтов

Напившись кофе, Дидье Хильхорст подошел к «Версии один» как к упражнению на создание запоминающейся торговой марки. Он применил ту же процедуру, что и для любого другого творческого произведения, приняв во внимание потенциальную аудиторию сайта Zen Garden (веб-дизайнеры), требования (корректный код, изобретательный дизайн) и конкуренцию (другие сайты с отличным дизайном).

Работая над «Версией один», он ставил себе задачу создать новый слоган для Zen Garden. Круговой логотип должен напоминать ромашку, но при этом остаться достаточно абстрактным, чтобы вписаться в окружающий дизайн (рис. 1). Смелые оттенки зеленого и оранжевого завладевают вниманием зрителя, а более мягкая серая гамма и однотонные области оставляют место, на котором есть где отдохнуть взгляду, скользящему сверху вниз.



Рис. 1 ✦ Стилизованный логотип Zen Garden

Размер шрифта

Хотя печатная страница не передает точно масштаба и пропорций, характерных для компьютерного экрана, но, возможно, вы заметили, что шрифты выглядят мелковато. А взглянув на таблицу стилей, вы поймете, почему это так:

```
p {
  font-size: 11px;
  text-align: justify;
}
#quickSummary p.p1 {
  font-size: 9px;
  background: transparent;
  color: #999;
  text-align: left;
}
```

Это означает, что самый крупный шрифт (не считая тех, что в изображениях) по высоте не превышает 11 пикселей, а местами используется даже шрифт высотой 9 пикселей. Дизайнеры любят мелкие шрифты, поскольку при большом кегле используемые в Web шрифты выглядят непрофессионально. (Особенно этим страдает шрифт Verdana с кеглем больше 12 px.) Кроме того, мелкий шрифт позволяет уместить в заданной области больше информации. Хорошо, если бы все мы обладали зрением дизайнера, но, увы, это не так.

Средства, предоставляемые браузером

К счастью, у посетителя сайта есть кое-какие возможности. Все основные браузеры позволяют изменять размер шрифта, делая его больше или меньше по желанию пользователя. В большинстве браузеров средства масштабирования находятся в меню **View** (Вид) или активируются при одновременном нажатии клавиш **Ctrl** + (плюс) и **Ctrl** – (минус).

Если бы все шло, как задумано, то здесь можно было бы и закончить главу, заполнив оставшиеся страницы шутками и анекдотами. Увы, имеется проблема: в Internet Explorer для Windows единица измерения px не масштабируется.

Относительные и абсолютные единицы измерения

Абсолютные единицы измерения длины имеют предопределенный смысл: `cm`, `in` и `pt` - это соответственно сантиметры, дюйм и пункт. Величина каждой из них известна, и бессмысленно доказывать, что 2,5 cm равны полутора футам.

С другой стороны, использование относительных единиц - `%`, `em`, `larger`, `smaller` - предполагает, что между размерами существует некоторое соотношение. Ясно, что `2em` и 200% в два раза больше, чем `1em` и `100%`, но что такое вообще `1em`? И 100 процентов *от чего*? Для каждой из этих единиц есть своя точка отсчета, и тут-то и начинаются различия: `1em` может быть равно и 3 дюймам, и 0,55 см, и еще много чему. Когда в таблице стилей применяются относительные единицы измерения, то устанавливается некий опорный размер, соответствующий `1em` или `100%`, и размеры шрифтов вычисляются *пропорционально*. Для заголовков может быть задан шрифт размером `150%` или `2,2em`, для подписей – `0,8em` или `75%`. Опорный размер можно задать явно или оставить на усмотрение браузера, но в любом случае пользователь сможет его изменить, в результате чего все шрифты будут отмасштабированы соответственно. Именно возможность управления со стороны пользователя и есть основное достоинство относительных единиц. Посмотрите, как выглядит «Версия один» в Windows XP при отключенной функции Clear Type, и вы поймете, почему эта возможность так важна.

Есть еще и третий вид единиц измерения. Согласно спецификации CSS, ключевые слова `xx-small`, `medium`, `x-large` и некоторые другие тоже являются абсолютными единицами. Однако все не так замечательно, как кажется; ни один метод не работает, как задумано.

Пиксели и проблемы

Согласно спецификации CSS, единица измерения `px` на самом деле является относительной. На этот счет существуют длинные и туманные рассуждения, связанные с длиной руки и углом зрения, но если в двух словах, то на 15-дюймовом мониторе с разрешающей

ПРИМЕЧАНИЕ

Если вы пользуетесь современной ОС, в которой есть механизм сглаживания шрифтов, например Mac OS X или Windows XP (в последнем случае нужно вручную включить функцию ClearType), то сможете без особого труда рассмотреть шрифты, использованные в «Версии один» (рис. 2 и 3). Если же сглаживание шрифтов не реализовано, то для чтения текста придется приложить усилия (рис. 4). Некоторые шрифты на удивление хорошо выглядят при малом кегле (примером может служить Verdana), тогда как другие становятся нечитаемыми.

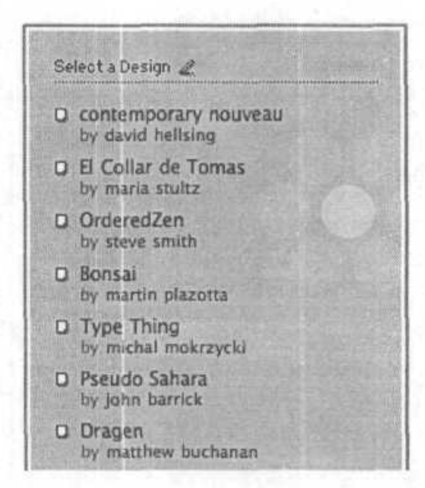


Рис. 2 ✦ Из-за сглаживания шрифтов в Mac OS X текст «Версии один» выглядит хоть и мелким, но вполне читаемым

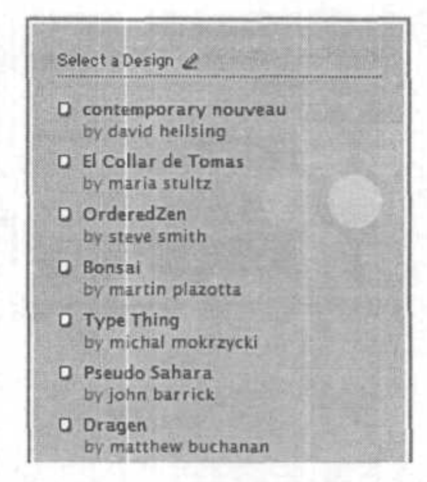


Рис. 3 ✦ Функция ClearType в Windows XP также справляется с задачей, хотя текст выглядит еще мельче



Рис. 4 ✦ В Windows XP при отключенной функции Clear Type читать «Версию один» трудно

Text as default; no size at all.	Text as default; no size at all.
Text as default with base 100%.	Text as default with base 100%.
Text as 12 pixels.	Text as 12 pixels.
Text as 12 pixels with base 100%.	Text as 12 pixels with base 100%.
Text as 80%	Text as 80%
Text as 80% with base 100%	Text as 80% with base 100%
Text as 0.8em.	Text as 0.8em.
Text as 0.8em with base 100%	Text as 0.8em with base 100%
Text as keyword small.	Text as keyword small.
Text as keyword small, ALA style.	Text as keyword small, ALA style.
Text using Matt Round's script.	Text using Matt Round's script.
Text using Matt Round's script without base 100%	Text using Matt Round's script without base 100%
Text as default; no size at all.	Text as default; no size at all.
Text as default with base 100%.	Text as default with base 100%.
Text as 12 pixels.	Text as 12 pixels.
Text as 12 pixels with base 100%.	Text as 12 pixels with base 100%.
Text as 80%	Text as 80%
Text as 80% with base 100%	Text as 80% with base 100%
Text as 0.8em.	Text as 0.8em.
Text as 0.8em with base 100%	Text as 0.8em with base 100%

Рис. 5 ✦ Некоторые из многочисленных тестов Оуэна Бриггса, демонстрирующие различия в определении размеров в разных браузерах

СОВЕТ

Подробнее о тестах Оуэна Бриггса см. статью «Text Sizing» по адресу http://thenoodleincident.com/tutorials/box_lesson/font/index.html.

способностью 1280×960 пикселей текст выглядит гораздо мельче, чем на 21-дюймовом мониторе с разрешающей способностью 800×600. Поскольку во втором случае шрифт с кеглем `10px` вполне различим, а в первом – вряд ли, то эту единицу измерения нельзя считать абсолютной в соответствии с данным ранее определением.

Но если `px` - относительная единица, то разве нельзя ее применить для точного контроля над опорным размером шрифтов и позволить пользователю масштабировать заданные с ее помощью шрифты по собственному желанию? Оно, конечно, так, только вот... Internet Explorer для Windows не умеет масштабировать единицу `px`, так что на практике она становится абсолютной. Но, быть может, тогда `em` или `%` решат все проблемы? Не торопитесь. Относительные единицы измерения тоже не идеальны.

Точка отсчета

Напомним, что относительные единицы измеряются относительно опорного размера. Но как его задать? Положившись на умалчиваемые значения в браузере. В большинстве браузеров в качестве опорного размера берется 16 пикселей – для многих людей это много.

Несогласованность и округление

Отдать выбор точки отсчета на усмотрение браузера, может быть, и хорошо, если только вам не нужно, чтобы в разных браузерах дизайн выглядел примерно одинаково. Хотя в большинстве браузеров и действует соглашение о 16 пикселях, но в разных операционных системах и старых версиях браузеров различия ошеломляющие. Несколько лет назад дизайнер Оуэн Бриггс (Owen Briggs) провел серию тестов и, собрав сотни снимков с экрана, доказал, что согласованность между относительными единицами измерения – не более чем несбыточная мечта (рис. 5). То же относится и к ключевым словам, определенным в спецификации CSS. Из-за этих расхождений долгое время считалось, что относительными единицами пользоваться нельзя; `1em` могло равняться чему угодно в диапазоне от 12 до 16 пикселей.

Различны не только точки отсчета, но и результаты масштабирования. Если `1em` равен 16 пикселям, то чему равен `0,9em`? В одном браузере 14,4 пикселей округляется до 14, в другом – до 15.

Сложные проценты

Пользуясь относительными единицами измерения, легко загнать себя в угол. Рассмотрим следующий код:

```
<div id="benefits">
  <h3>Benefits</h3>
  <p>Why participate? For recognition,
inspiration, and a resource we
  can all refer to when making the case for
CSS-based design. This is
  sorely needed, even today.</p>
  </div>
```

Если применить к нему показанную ниже таблицу стилей, то каков будет размер текста внутри абзаца в предположении, что 100% – это 16 пикселей?

```
#benefits {
  font-size: 75%;
 }
 p {
  font-size: 75%;
 }
```

Что у вас получилось: 9 или 12 пикселей? Правильный ответ – 9. Первое правило уменьшает размер шрифта внутри раздела `#benefits` на 75%, то есть до 12 пикселей. Второе правило наследует это значение и уменьшает его еще на 75%, в результате получается шрифт высотой 9 пикселей.

Эта схема вычисления сложных процентов приводит в недоумение, когда применяется неосознанно и без проверки; сделайте несколько таких итераций – и текст станет вовсе нечитаемым. Особенно от этого страдают старые браузеры, например Internet Explorer для Macintosh: с каждым применением относительных единиц к вложенным элементам текст становится все меньше и меньше (рис. 6 и 7).

Рис. 6 ✦ В современном браузере размер текста кажется приемлемым...

Рис. 7 ✦ ...а в более старом из-за перемножения коэффициентов текст плохо читается

СОВЕТ

В сети есть много рекомендаций, основанных на здравом смысле, например на сайтах MaxDesign (www.maxdesign.com.au/presentation/relative), css-discuss (http://css-discuss.incutio.com/?page=FontSize) и Dive Into Accessibility (http://diveintoaccessibility.org/day_26_using_relative_font_size.html).

Разумное масштабирование шрифтов

Итак, все методы задания размера шрифта порождают те или иные проблемы. Так что же делать бедному веб-дизайнеру? Похоже, задание размера в пикселях – самый простой выход для дизайнера, зато доставляет максимум неприятностей пользователю. Существует три надежных способа, и у каждого из них есть плюсы и минусы.

Метод em

После того как Оуэн Бриттс представил миру свои снимки с экрана, он обнаружил, что при задании размера шрифта с помощью процента самых очевидных ошибок округления удается избежать. Если в элементе body указать процентное значение, а во всех остальных местах единицу em, то размер вычисляется приблизительно одинаково во всех браузерах:

```
body {
 font-size: 76%;
}
p {
 font-size: 1em;
}
```

Бриттс предлагает именно такие значения и рекомендует не употреблять для масштабирования текста на странице коэффициент ниже 1em. В результате шрифт внутри элемента p будет иметь высоту примерно 12 пикселей при условии, что по умолчанию в браузере предполагается кегль 16 пикселей. Для большей части людей это

достаточно комфортно, а в случае чего относительность единиц измерения позволит изменить масштаб. Указанный метод подробно описан на личном сайте Бриггса в статье «Sane CSS sizes» (www.thenoodleincident.com/tutorials/typography/ index.html). Ахиллесовой пятой этой техники является предположение о том, что браузер по умолчанию принимает кегель 16 пикселей. Для большинства браузеров так оно и есть, но если пользователь изменит стандартное значение, то это отразится и на масштабировании.

Метод на основе ключевых слов

Самые серьезные проблемы при задании размера шрифта с помощью ключевых слов возникают в браузере Internet Explorer 5 для Windows. Если забыть о нем, то этот метод дает достаточно согласованные результаты и в большинстве остальных браузеров работает надежно. Так что, отфильтровав Internet Explorer 5 для Windows с помощью трюка, связанного с моделью прямоугольных областей (Box Model Hack), ключевыми словами пользоваться все-таки можно:

```
body {
 font-size: x-small;
 voice-family: "\"}\"";
 voice-family: inherit;
 font-size: small;
}
html>body {
 font-size: small;
}
```

Предполагается, что при переходе от одного ключевого слова к соседнему действует масштабный коэффициент 1,2. Таким образом если среднее значение соответствует 16 пикселям, то приведенное выше объявление даст размер примерно 13 пикселей.

ПРИМЕЧАНИЕ

Трюк с моделью прямоугольных областей в CSS позволяет применить специальный стиль только в Internet Explorer 5 для Windows. Он был придуман для обхода дефекта реализации этой модели, характерной исключительно для данного браузера, но применяется и в других ситуациях, в том числе для задания размера шрифтов. Подробнее см. на странице http://tantek.com/CSS/Examples/ boxmodelhack.html.

ПРИМЕЧАНИЕ

Детальное описание того, что именно делает приведенный выше код, можно найти на странице http://diveintoaccessibility.org/day_26_using_relative_font_size.html.

ПРИМЕЧАНИЕ

Описание метода Пола Соудена приведено в статье «Alternative Style: Working with Alternate Style Sheets»(www.alistapart.com/articles/alternate).

Метод переключения таблицы стилей

Для точного контроля над экранными шрифтами лучше всего пользоваться единицей измерения px. Вообще-то полагаться на нее опасно, но существует способ обойти ошибку в Internet Explorer, создав вторую таблицу стилей с большим размером шрифта.

Методы переключения таблиц стилей встроены в некоторые браузеры, но не в Internet Explorer для Windows. Поэтому были разработаны программные способы восполнения этого недостатка, с помощью которых пользователь может взять управление в свои руки. Наиболее известен оригинальный метод Пола Соудена (Paul Sowden), опубликованный на сайте A List Apart. С его помощью можно модифицировать не только шрифты, он доказал свою эффективность в построении средств настройки сайта с помощью экранных элементов управления, добавленных дизайнером.

Так какой метод выбрать?

На каком же из трех этих методов лучше остановиться при работе над своим проектом? Дело вкуса. У посетителей, заходящих на сайт, зрение может оказаться похуже, чем у дизайнера Дидье Хильхорста, но если вы тщательно подойдете к выбору метода задания размеров шрифта, то они не станут на это сетовать.

Дизайнер **Майкл Пик** (Michael Pick)
www.csszengarden.com/009

Живым или мертвым

Изящество и притяжение в типографском дизайне

Размышляя над тем, что представить на сайт CSS Zen Garden, Майкл Пик решил сделать «что-нибудь забавное и немного странное». Так и появилась работа «Живым или мертвым».

Идея Пика состояла в том, чтобы противопоставить дикий Запад миру японского самурая, показав занимательное историческое различие. На тему дикого Запада намекает типография в стиле вестернов с различными специальными символами. С точки зрения типографских особенностей получился очень интересный результат, проливающий свет на проблему, как надо использовать шрифты в дизайне.



A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED VISUALLY THROUGH CSS-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.

DOWNLOAD THE SAMPLE HTML FILE AND CSS FILE

★★ SELECT A DESIGN ★★

THE FINAL ENDING BY RAY HENRY
CONTEMPORARY NOUVEAU BY DAVID HELLSING
EL COLLAR DE TOMAS BY MARIA STULTZ
ORDERREIZED BY STEVE SMITH
BONSAI BY MARTIN PLAZOTTA
TYPE THING BY MICHAL MOKRZYCKI
PSEUDO SAHARA BY JOHN BARRICK
DRAGEN BY MATTHEW BUCHANAN

★★ ARCHIVES ★★

NEXT DESIGNS »
VIEW ALL DESIGNS

★★ RESOURCES ★★

VIEW THIS DESIGN'S CSS
CSS RESOURCES
FAQ
SUBMIT A DESIGN
TRANSLATIONS

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT ?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of *those able to create beauty from structure. To date, most examples of* neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

AN INVITATION TO PARTICIPATE

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the **CSS Resource Guide** for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample **html file** and **css file** to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. **Send us a link** to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

THE BENEFITS OF PARTICIPATION

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

TO BE REQUIRED OF THE PARTICIPANTS

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the **Resources** page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see **submission guidelines**), but we ask you release your CSS under a Creative Commons license identical to the **one on this site** so that others may learn from your work.

Bandwidth graciously donated by **DreamFire Studios**. Coming soon: Zen Garden, the book.

xhtml css cc 508 aaa



Рис. 1 ✦ Специальные шрифты обычно создаются для какой-то одной темы



Рис. 2 ✦ Декоративный логотип в дизайне «Живым или мертвым»

★★ SELECT A DESIGN ★★

Рис. 3 ✦ Заголовки в списках ссылок стилизованы с использованием звезд и специального символа-разделителя



THE ROAD TO ENLIGHTENMENT

Рис. 4 ✦ В верхнем заголовке изображены самураи и бонсай

SO WHAT IS THIS ABOUT ?

Рис. 5 ✦ Дополнительные заголовки выдержаны в стиле вестерна

Декоративные символы

Один из самых любопытных визуальных аспектов работы «Живым или мертвым» является использование декоративных типографских символов, получивших собирательное название *dingbats* («штучки-дрючки»). Традиционно это были цветочки или элементы орнамента, а в цифровом мире они подбираются по темам, например:

- ✦ бизнес;
- ✦ спорт;
- ✦ животные;
- ✦ природа;
- ✦ разное.

На рис. 1 показаны примеры символов на пять разных тем.

Пик сумел подчеркнуть тему вестерна в своей работе специальными символами и графическими украшениями, поэтому дизайн получился изощренным и искусным.

Прежде всего бросается в глаза заголовок (рис. 2). Шрифт, вызывающий ассоциации с диким Западом, сочетается с изображением пагоды, которая помещена в овальную область, обрамленную причудливыми завитушками. В результате получается тематический контраст между диким Западом и древней Японией. Пятиконечная звезда, то есть звезда шерифа, хорошо знакома по фильмам-вестернам. В списках ссылок заголовки обрамлены такими звездами и отделены специальным шрифтовым символом (рис. 3).

Перемещаясь вниз по основному тексту, мы обнаруживаем дополнительные украшения. Во-первых, в верхней части видны изображения фигур самураев и дерево-бонсай (рис. 4).

В последующих заголовках доминирует тема дикого Запада, подчеркнутая стилизованным изображением указающего перста (рис. 5).

Для усиления впечатления применены еще два декоративных элемента. Вдоль верхнего и нижнего края размещен красивый узор, напоминающий о банкнотах и сертификатах акций (рис. 6). И наконец, почтовый штемпель с доменным именем mikepick.com служит тонким украшением с толикой юмора (рис. 7).

Не будь на странице специальных символов и графических украшений, осознать тематический контраст было бы непросто. На этом примере видно, как дизайнер может с помощью удачно размещенных декоративных элементов подчеркнуть свою идею.

Типографская верстка

Говоря о верстке, мы прежде всего имеем в виду размещение различных элементов на странице. На эту тему существует много как научной информации, так и всяческих шуток.

Оптическое восприятие

Человеческий глаз воспринимает текст очень специфически. Типографы советуют дизайнерам строить свои работы в соответствии с механизмами человеческого зрения и не пытаться делать что-то, противоречащее естественной реакции на слова. Единственное исключение из этого правила – когда дизайнер сознательно хочет вызвать необычную реакцию. Во всех остальных случаях важно учитывать оптическое восприятие.

В языках, читаемых слева направо, взгляд следует в определенном направлении с самого начала страницы. Это верно и для печатных, и для веб-страниц.

В типографии преобладает теория, что взгляд естественно тяготеет к левому верхнему углу, который называют *главной оптической областью.* Затем взгляд движется вправо, спускается вниз на следующую строку и так далее до конца страницы. В конечном итоге взгляд упирается в так называемый *финальный анкер* – самую нижнюю точку в дизайне. В идеале, в макете, предназначенном для практического, а не экспериментального применения, такой характер оптического восприятия должен учитываться, и разрушать его не следует (рис. 8).

Использование притяжения для направления взгляда

Процесс направления взгляда вниз называется *притяжением* (gravity). Как сила тяжести физически тянет нас вниз, так и взгляд притягивается к области естественного завершения дизайна.

Зная это, вы не станете удивляться тому, что логотип обычно помещают в главной оптической области (левом верхнем углу), а текст и элементы дизайна, создающие финальный анкер, располагают внизу. По пути из одной точки в другую встречаются «горячие точки» – обычно там, где взгляд останавливается перед

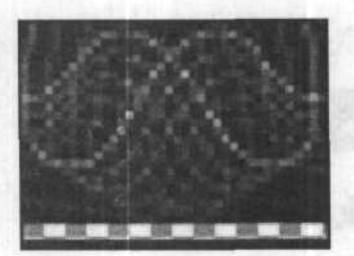


Рис. 6 ✦ Этот повторяющийся узор создает некую интригу (изображение увеличено)



Рис. 7 ✦ Почтовый штемпель выступает в роли водяного знака (изображение затемнено)

ПРИМЕЧАНИЕ

Конечно, в языках, читаемых не слева направо, оптическое восприятие иное. В иврите или арабском, которые читаются справа налево, логично предполагать прямо противоположное направление взгляда: из правого верхнего угла поперек и вниз.



Рис. 8 ✦ Схема оптического восприятия при чтении страницы

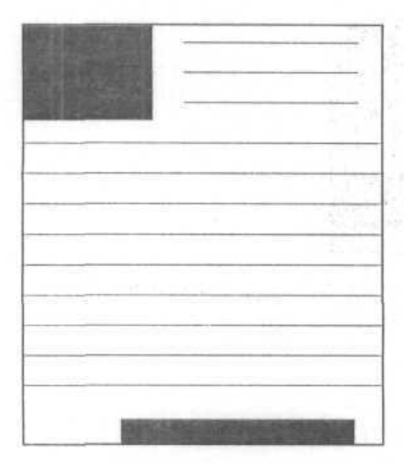


Рис. 9 ✦ Шаблон верстки, не противоречащей оптическому восприятию

SO WHAT IS THIS ABOUT ?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

Рис. 10 ✦ В работе «Живым или мертвым» заголовки не содержат иных знаков препинания, кроме единственного вопросительного знака. В сочетании с указующим перстом они помогают направить взгляд читателя к следующему фрагменту текста

тем, как перейти в левую часть страницы. Шаблон, идеально соответствующий такой модели, представлен на рис. 9. Работа «Живым или мертвым» дает прекрасный пример дизайна, в котором взгляд не встречает препятствий естественному движению.

Как избежать непонимания

Другая проблема, возникающая в типографии, – как добиться того, чтобы человек правильно понял смысл выраженной словами информации. Как и раньше, экспериментальный дизайн может представлять исключение из правила: скажем, сообщение может быть заложено в эмоциональной реакции на некий образ, а не в словесном содержании.

Заголовки

Разумеется, заголовки призваны отделить друг от друга разделы документов. В работе «Живым или мертвым» мы видим, что заголовок может еще и нести дополнительный смысл, усиливая впечатление от дизайна.

Вот некоторые рекомендации по эффективному использованию заголовков:

- ✦ если вы используете в заголовках цвет, то это должен быть самый заметный цвет в палитре;
- ✦ цвет заголовка должен быть энергичным. Исследования подтверждают, что светлые заголовки не оказывают такого воздействия, как темные. Чем ближе цвет заголовка к черному, тем сильнее он воздействуют зрителя;
- ✦ чем значимее заголовок, тем он должен быть заметнее;
- ✦ слегка зауженный шрифт заголовка делает его более заметным. Иногда рекомендуют устанавливать плотность на уровне 70–90%;
- ✦ полезны бывают шрифтовые контрасты. В работе «Живым или мертвым» все заголовки набраны заглавными буквами, что выделяет их на фоне прочего текста;
- ✦ заголовки должны быть короткими. Слишком длинный заголовок выглядит неприятно и может быть непонятен читателю.

Еще один интересный фактор, который нужно учитывать в дизайне заголовков, – это воздействие знаков препинания, в особенности точек. Вообще говоря, следует избегать точек в конце заголовков. Точка символизирует завершение мысли, тогда как смысл заголовка – начать новую мысль (рис. 10).

Основной текст

Основной текст занимает большую часть страницы. Предполагается, что посетитель будет его читать или хотя бы бегло просмотрит. Но следование сложившейся типографской практике может уменьшить доходчивость сообщения.

Для эффективной работы с основным текстом придерживайтесь следующих рекомендаций:

- ✦ Старайтесь, чтобы строка состояла примерно из 60 символов. В текучем мире веб-дизайна это не всегда легко обеспечить, особенно из-за не всегда корректной реализации свойств CSS, управляющих длиной строки, к примеру, `min-width` и `max-width`. Однако это следует иметь в виду, даже при текучем дизайне. Если строки слишком длинные, то их смысл становится труднее понимать. Более того, человек вообще может отказаться от чтения текста!
- ✦ Верно и обратное. Восприятие смысла падает и при слишком коротких строках, порядка 20 символов. Просто трудно читать длинные и узкие фрагменты текста. Короткие строки лучше применять для акцентирования внимания.
- ✦ В англоязычных текстах принято считать, что абзац должен состоять по меньшей мере из трех-четырех предложений. В Web длинные абзацы не приветствуются, считается, что короткие абзацы, содержащие действительно важную информацию, больше подходят для среды, в которой одного щелчка достаточно, чтобы покинуть страницу.
- ✦ Публикуя длинные документы, старайтесь, чтобы абзацы были насыщенными и короткими, разбивайте документ на разделы с помощью заголовков. Это, безусловно, способствует улучшению смыслового восприятия, поскольку читателю становится легче ориентироваться в документе.
- ✦ Избегайте слишком широких и слишком узких межстрочных интервалов. Если абзац слишком близко или слишком далеко отстоит от предшествующего и последующего текста, восприятие его смысла ухудшается. Мозг просто перестает обрабатывать информацию либо из-за перегрузки, либо считая, что сообщение закончилось.
- ✦ По большей части старайтесь не использовать разные цвета в основном тексте. Как и в случае заголовков, раскраска текста иногда приносит больше вреда, чем пользы. Применяйте цвет только для акцентирования внимания и для выделения ссылок. В работе «Живым или мертвым» даже в ссылках

Download the sample **html file** and **css file** to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. **Send us a link** to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Рис. 11 ✦ Вариация цвета для ссылок внутри основного текста (дизайн «Живым или мертвым»)

применяется только небольшая вариация оттенка, да и то лишь в тех, что находятся в основном тексте (рис. 11).

Акцентированный текст

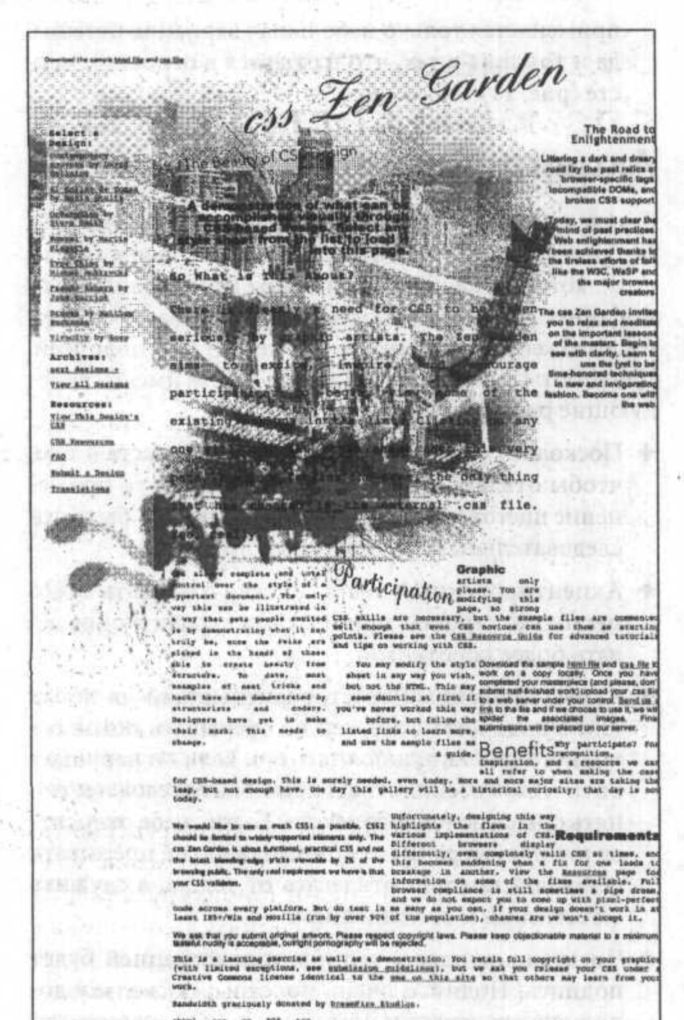
С акцентированным текстом можно позволить чуть больше вольности, чем с заголовками и основным текстом, которые наиболее критичны для уяснения сути информации. Акцентированным называется текст, расположенный по бокам, в виде цитат и подписей. Для работы с акцентированным текстом имеются следующие рекомендации:

- ✦ Поскольку смысл акцентированного текста в том, чтобы отвлечь внимание от основного, то применение цветов даже поощряется. Главное – быть последовательным в использовании цветов.
- ✦ Акцентированный текст должен занимать всего несколько строчек, причем сами строки лучше делать более короткими.
- ✦ Акцентированный текст, размещенный на полях или в виде цитат, должен не прерывать поток основного текста, а дополнять его. Если он нарушает оптическое восприятие, то становится сложнее понять смысл основного текста. То же самое относится и к иллюстрациям – они должны не прерывать основной поток и отвлекать от текста, а служить естественным дополнением.
- ✦ Читатель ожидает, что под иллюстрацией будет подпись. Подписи очень полезны, поскольку дополняют контекст и помогают лучше усвоить материал.

Решая вопрос о том, сколько текста может быть на одной странице, следует сообразовываться с практичностью. По большей части, люди не любят щелкать по ссылке «Далее» чаще, чем необходимо. Поэтому нужно соблюсти тонкий баланс: оставить на странице не слишком мало и не слишком много текста.

Продуманные шрифты, изящный дизайн

Дизайн работы «Живым или мертвым» выглядит изящно благодаря тому вниманию, которое Майкл Пик уделил мелким деталям и удачно подобранным украшениям в виде символов из фантазийных шрифтов, с помощью которых высвечены интересные грани. Добавьте к этому твердое следование типографским принципам, и вы поймете, что в основе выразительности работы лежит тонкая, но ясно ощущаемая простота.



Дизайнер **Дэйв Ши** (Dave Shea)
www.csszengarden.com/005

Жажда крови

Исследование возможностей форматирования

Иногда для продвижения вперед приходится пользоваться воспоминаниями из прошлого. Работая над хаотической «Жаждой крови», Дэйв Ши обратился к футуристическому искусству начала двадцатого века. Была поставлена цель соединить элементы прошлого с современной технологией.

Основу образного ряда составили фотографии из личного архива Ши. В сочетании с несколькими маскированными слоями получилась двухцветная репродукция, напоминающая шелковые трафареты, которые часто применялись для графического дизайна в прошлом. В очень ограниченной цветовой палитре используются только черный, красный и белый цвета. Чтобы подчеркнуть резкость, он оставил на всех изображениях контуры, отказавшись от сглаживания неровностей. А воспользовавшись сглаживанием переходов (dithering) в формате GIF, он сумел тонко намекнуть на возможности современной технологии.

Задание настроя с помощью типографских эффектов

Беспорядочность, ощущаемая в «Жажде крови», специально подчеркнута организацией текста; абзацы позиционированы и отформатированы по-разному, что создает впечатление хаоса. Применение различных гарнитур широко практиковалось дизайнерами-футуристами, при этом текст специально размещался так, чтобы создать максимальный зрительный дискомфорт (рис. 1).



Рис. 1 ✦ Пример футуристического дизайна, художник Ф. Т. Маринетти «Les mots en liberte futurists» (1919)

> **ПРИМЕЧАНИЕ**
>
> В типографии цветом называют не оттенок краски, а зрительно воспринимаемый общий тон фрагмента текста. Соотношение между формой литеры и поверхностью, на которой она напечатана, зависит прежде всего от контраста; без контраста текст будет нечитаемым. Побочным продуктом этого соотношения является тот факт, что промежутки – между штрихами литеры, между буквами внутри слова, между словами в строке и между строчками – зависят от толщины шрифта и величины межбуквенных и междусловных интервалов.

Чтобы добиться чего-то необычного в «Жажде крови», понадобилось разместить абзацы так, чтобы они обтекали друг друга и заполняли страницу беспорядочным образом. С точки зрения удобства чтения такая организация материала далека от идеала, но, возможно, вы заметили интересный эффект: межбуквенные интервалы и плотности шрифтов различаются, из-за чего разные параграфы имеют разные оттенки. На рис. 2, где изображение слегка размыто, чтобы скрыть отдельные литеры, влияние плотности шрифта отчетливо видно.

В данной работе наибольший интерес представляет случайно меняющийся цвет шрифта. Изменяя межстрочные интервалы, поля и гарнитуру шрифта, можно добиться разного восприятия сообщения. Какой из фрагментов, представленных на рис. 3, легче читать? Решив поиграть типографскими эффектами в собственном проекте, не упускайте из виду эти соображения.

В большинстве случаев цель дизайна – последовательно выдержать один и тот же цвет во всем основном тексте. В CSS не так много средств, позволяющих решить эту задачу. Но, хотя забывать о ней не следует, есть и более важные вещи – к примеру, четкость и длина строк.

Возможности форматирования

В «Жажде крови» широко используются различные средства форматирования текста, имеющиеся в CSS, помимо таких основных, как задание размера и гарнитуры шрифта.

В спецификации CSS, а равно в многочисленных книгах и руководствах по этой теме, вы найдете техническую информацию по каждому свойству, но ничего по поводу их применения в практическом дизайне. Некоторые включенные в CSS свойства глубоко корнями уходят в основные принципы графического дизайна, но из технического руководства вы об этом никогда не узнаете.

Ниже будут описаны наиболее распространенные свойства, относящиеся к форматированию, их происхождение и возможные способы применения. Некоторые свойства (к примеру, `font-family`, `font-size`, `font-style` и `font-weight`) опущены, так как подробно рассматриваются в других разделах книги.

Применение стилей к тексту

Описываемые в этом разделе свойства позволяют задавать формат вплоть до уровня отдельного символа. В большинстве случаев у шрифта *должен быть* предопределенный вариант, соответствующий выбранному стилю. Знакомство со списком шрифтов, установленных на вашей машине, может пролить свет на то, какие варианты есть у каждого шрифта.

Свойство font-variant

У этого свойства одно-единственное назначение – отобразить текст маленькими заглавными буквами. Высота первой буквы каждого слова остается обычной, а весь остальной текст печатается маленькими заглавными буквами.

Свойство может принимать одно из двух значений: `normal` и `small-caps`. У большинства шрифтов варианта с маленькими заглавными буквами нет, поэтому браузер просто уменьшает размер обычных заглавных букв. Поэтому сделанное выше замечание о предопределенных вариантах шрифта к свойству `font-variant` не относится.

Пример использования `font-variant`:

```
#quickSummary p {
 font-variant: small-caps;
}
```

Как выглядит результат:

The Beauty of *CSS* Design

A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED VISUALLY THROUGH *CSS*-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.

DOWNLOAD THE SAMPLE HTML FILE AND CSS FILE

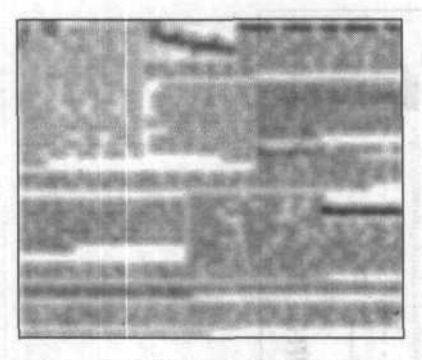


Рис. 2 ✦ После размывания становится видно, что у блоков текста разные оттенки



Рис. 3 ✦ Примеры разного цвета шрифта

ПРИМЕЧАНИЕ

Цветной вариант рис. 2 и 3 см. на вкладке.

СОВЕТ

В спецификации CSS приведено даже больше деталей о каждом свойстве стилизации шрифтов; см. раздел «15 Fonts» на странице www.w3.org/TR/REC-CSS2/ fonts.html#font-styling.

СОВЕТ

У каждого из обсуждаемых свойств есть также значение `inherit`. Оно заставляет браузер применить то же значение, что действует для родительского элемента. Это относится ко всем свойствам CSS, поэтому мы не будем упоминать его явно.

Свойство text-transform

Как и `font-variant`, свойство `text-transform` позволяет более точно управлять преобразованием строчных букв в заглавные. Оно может принимать следующие значения: `capitalize`, `uppercase`, `lowercase` и `none`.

Значение `capitalize` преобразует в верхний регистр первую букву каждого слова, оставляя все прочие без изменения; значение `uppercase` преобразует все буквы в верхний регистр, `lowercase` переводит все буквы в нижний регистр, а `none` не имеет видимого эффекта.

Пример использования `text-transform`:

```
#quickSummary p {
 text-transform: uppercase;
}
```

Как выглядит результат:

The Beauty of *CSS* Design

A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED VISUALLY THROUGH CSS-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.

Свойство text-decoration

С самого момента зарождения Всемирной паутины гиперссылки подчеркивались, свойство же `text-decoration` позволяет управлять подчеркиванием и другими эффектами не только для ссылок, но и для любого другого текста. Допустимые значения: `none`, `underline`, `overline`, `line-through` и `blink`. Значение `none` не имеет видимого эффекта, `underline` добавляет линию снизу (подчеркивание), `overline` - линию сверху (надчеркивание), `line-through` – линию поперек текста (перечеркивание), а `blink` вызывает мигание.

Наверное, `text-decoration` является самым противоречивым из всех свойств форматирования текста, поскольку может вызывать раздражение у читателя. Чтобы избежать этого, придерживайтесь следующих рекомендаций:

- `none` –подразумевается для большинства элементов; значение имеет смысл задавать явно, только если оно отлично от `none`.
- `underline` – применяйте только для гиперссылок. Если текст подчеркнут, но не является гиперссылкой, то пользователь будет введен в заблуждение. Сколько раз вы сами пытались щелкнуть по

подчеркнутому тексту, не являющемуся ссылкой? В типографии у подчеркивания другой, более традиционный смысл: так, например, выделяют названия книг, но наличие гиперссылок внесло свои коррективы;

- overline – в математике надчеркивание применяется для обозначения средних величин, но больше оно почти никогда не используется;
- line-through – применяйте перечеркивание только для выделения текста, который потерял актуальность в контексте объемлющей страницы и вообще-то заслуживает изъятия, но по какой-то причине все же оставлен. В идеале любой вычеркнутый текст должен сопровождаться более новым; хорошим примером применения могут служить системы учета исправлений и ведения версий;
- blink – если у вас нет абсолютной необходимости имитировать мигающий таймер видеомагнитофона, считайте, что значения blink не существует. На заре развития Web тегом <blink> чрезмерно злоупотребляли, а сегодня не каждый браузер поддерживает его эквивалент в CSS.

Пример использования text-transform:

```
#quickSummary p {
 text-decoration: line-through;
 }
```

Как выглядит результат:

The Beauty of *CSS* Design

~~A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.~~

~~Download the sample html file and css file~~

Управление промежутками

Выше мы видели, как контраст между литерами и фоном влияет на восприятие текста. Промежутки – это мощный визуальный инструмент, а благодаря развитым возможностям форматирования в CSS, ими еще и легко управлять.

Свойство line-height

В прошлом типографы отделяли строки друг от друга полосками мягкого свинца (lead), отсюда и поныне существующий английский термин *leading* для обозначения межстрочного интервала, или интерлиньяжа.

CSS позволяет управлять величиной этого интервала с помощью свойства `line-height`. Допустимо задавать его значение в абсолютных и относительных единицах, указывая длину, процентное отношение или ключевое слово `inherit`.

Отметим, что для значения свойства `line-height` явно задавать единицу измерения необязательно. Значение `1.2` – это то же самое, что `1.2em` или `120%`; если единица не задана, то считается, что к тексту применяется коэффициент масштабирования, а не абсолютный размер.

Пример использования `line-height`:

```
#quickSummary p {
 line-height: 0.8;
}
```

Как выглядит результат:

The Beauty of *CSS* Design

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

The Road to Enlightenment

Свойство letter-spacing

Межбуквенный интервал можно регулировать с помощью свойства letter-spacing. Оно может принимать значение normal или задаваться в виде длины промежутка, добавляемого к стандартному расстоянию между буквами.

ПРИМЕЧАНИЕ

Известный типограф Фредерик Гауди (Frederick Goudy) однажды сказал: «Всякий, кто изменяет межбуквенный интервал, способен украсть овцу». Пользуйтесь свойством `letter-spacing` с осторожностью, если не хотите быть обвиненными в угоне домашнего скота!

Из-за низкой разрешающей способности экрана изменять межбуквенный интервал в основном тексте можно только с довольно большой дискретностью, а это выглядит по-любительски. Поэтому наиболее разумно применять свойство `letter-spacing` в заголовках и небольших фрагментах текста.

Пример использования `letter-spacing`:

```
#quickSummary p {
  letter-spacing: 0.1em;
}
```

Как выглядит результат:

The Beauty of *CSS* Design

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

Свойство word-spacing

Как и letter-spacing, свойство word-spacing позволяет регулировать промежуток, только не между буквами, а между словами. Допустимые значения: normal, длина, процент или ключевое слово inherit.

На применение свойства word-spacing браузером влияют и другие факторы, в том числе выравнивание (задаваемое свойством text-align) и свойство white-space.

Пример использования word-spacing:

```
#quickSummary p {
 word-spacing: 1em;
}
```

Как выглядит результат:

The Beauty of *CSS* Design

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Свойство text-align

Это свойство позволяет отцентрировать или произвести выключку текста, а также выровнять его по правому или левому краю объемлющего элемента. Допустимые значения: left, right, center и justify.

Поскольку браузер выполняет выключку автоматически, то традиционные приемы предотвращения неприглядных пустот – к примеру, перенос слов – неприменимы. Выключка с помощью CSS часто приводит к ситуации, изображенной на рис. 4. Из-за отсутствия точного контроля мы рекомендуем ограничиться выравниванием по левому или правому краю. Центрирование может найти применение в заголовках и небольших фрагментах текста, например цитатах, но лучше не прибегать к нему для форматирования текста, длина которого превышает пару строк.

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Рис. 4 ✦ Отсутствие переносов в текстовых блоках приводит к большим межсловным пробелам

Пример использования `text-align`:

```
#quickSummary p {
  text-align: right;
}
```

Как выглядит результат:

The Beauty of *CSS* Design

A demonstration of what can be accomplished visually through *CSS*-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

Соблюдайте меру

Безоглядное применение произвольных комбинаций свойств шрифта, может быть, и неплохо смотрится в хаотическом дизайне «Жажды крови», но вообще-то тексты, публикуемые во Всемирной паутине, предназначены для чтения, а не для того, чтобы зритель восхищался мастерством художника. Если пользоваться предоставляемыми CSS возможностями для форматирования текста умеренно, то можно повысить четкость и доходчивость страницы.



Дизайнер **Дуглас Боумен** (Douglas Bowman)
www.csszengarden.com/017

Золотая середина

Сочетание метода и творческого подхода при создании типографского баланса

Работая над своим вкладом в копилку сайта CSS Zen Garden, дизайнер Дуглас Боумен понял, что хоть этот проект и нельзя назвать полномасштабным, но процесс разработки мало отличается от того, которому он следует при создании настоящих больших проектов.

«Золотая середина» демонстрирует, что типография в технологии веб-дизайна занимает такое же место, как и в других графических искусствах, и что дизайн – это сочетание вдохновения, технических деталей и процедуры.

Рис. 1 ✦ Экспериментирование с различными шрифтами в ходе работы над «Золотой серединой»

Выбор шрифта

Потратив время на размышления о содержании и идеологическом наполнении страницы, прикинув возможные варианты верстки, Боумен был готов с головой окунуться в вопросы типографии.

Может показаться, что выбор шрифта второстепенен по отношению к таким вещам, как использование графики и цвета, но опытные дизайнеры знают, что успех зависит от сочетания множества факторов. И впечатление от шрифта – один из них. На большинство людей шрифт воздействует на подсознательном уровне, зато это воздействие очень сильно. Шрифт может создать то или иное настроение, ассоциируясь с:

- юмором;
- стоицизмом;
- необузданностью;
- капризом;
- ностальгией.

Если вы можете добавить к этом списку еще какую-нибудь эмоцию, то весьма вероятно, что найдется шрифт или комбинация шрифтов, с помощью которых эту эмоцию можно выразить. Поэтому дизайнеру надлежит подойти к выбору шрифтов серьезно и ответственно.

Экспериментирование в Adobe Illustrator

Чтобы сделать правильный выбор, Боумен обратился к программе Adobe Illustrator. Объединив программное средство со своим глубоким знанием типографии, он вознамерился подобрать шрифты, в наибольшей степени отражающие то настроение, которое он собирался выразить в дизайне.

На рис. 1 представлены результаты первых экспериментов для работы «Золотая середина». Подыскать нечто, наилучшим образом подходящее для тонкого, юмористичного, оригинального дизайна, получилось не сразу. Заметьте, впрочем, что Боумен воспользовался исходными заголовками с сайта The CSS Zen

ПРИМЕЧАНИЕ

Illustrator – это программа, в которой используется векторная графика. Многие веб-дизайнеры сначала с ее помощью делают модель и подбирают шрифты, а затем переходят в Adobe Photoshop, чтобы преобразовать результат в форму, пригодную для Web (обе программы входят в состав комплекта Adobe Creative Suite). Компания Macromedia предлагает аналогичную программу Freehand в составе своей линейки продуктов Studio MX. Подробнее см. www.adobe.com/products/creativesuite/main.html и www.macromedia.com/software/studio.

Garden и The Beauty of CSS Design. Предварительный просмотр текста, набранного данным шрифтом, помогает оценить не только сам шрифт, но и то, как он смотрится в окружающем контексте.

В конечном итоге Боумен остановился на шрифте Morpheus, который можно бесплатно скачать с сайта Kiwi Media (www.kiwi-media.com/fonts.html) (рис. 2.). Его привлекли некоторые особенности данного шрифта, в частности:

- ✦ уникальная форма букв M и N;
- ✦ готические своды (особенно в буквах A и U);
- ✦ продленные буквы K и R;
- ✦ подобие таких литер, как S и строчная d;
- ✦ декоративные ромбы в строчной i, а также внутри D, O и Q.

Рис. 2 ✦ Шрифт Morpheus

Рис. 3 ✦ Заголовок из «Золотой середины», в котором использованы одновременно шрифты Morpheus и Helvetica

В поисках изюминки

Для создания некоей визуальной изюминки, сообщающей дополнительный интерес всему дизайну, Боумену нужен был еще контрастный шрифт. Он решил воспользоваться гарнитурой Helvetica. Это современный шрифт без засечек, хорошо контрастирующий с Morpheus. Таким образом был переброшен мост между готической и модерновой тональностью, что и придает работе такое очарование (рис. 3). К числу других контрастов можно отнести смешение в списках ссылок беззасечного шрифта Verdana со шрифтом Georgia, имеющим засечки (рис. 4). Чтобы ссылки состояли только из строчных букв, Боумен воспользовался свойством `text-transform`, указав значение `lowercase`:

```
#linklist #lselect a.c:link, #linklist
#lselect a.c:visited {
 display:inline;
 font-family:Georgia,Serif;
 font-weight:normal;
 color:#616623;
 background-color:transparent;
 text-transform:lowercase;
}
```

У свойства `text-transform` есть и другие значения: `capitalize` делает заглавными первые буквы слов, а `uppercase` переводит все буквы в верхний регистр.

Шрифт Georgia применяется в основном тексте (рис. 5) и в набранной курсивом преамбуле, это еще больше разнообразит шрифтовые аспекты «Золотой середины».

Рис. 4 ✦ Шрифты Verdana и Georgia в списке ссылок. Обратите внимание, что весь текст, набранный шрифтом Georgia, преобразован в нижний регистр, тогда как текст, отображаемый шрифтом Verdana, преобразованию не подвергался

so WHAT is this ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

the ROAD to enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in

Рис. 5 ✦ Обратите внимание на использование нормального и курсивного шрифта Georgia в основном тексте и преамбуле, а также на графические шрифты в заголовках

> **СОВЕТ**
>
> Конечно, никто не мешает вам вволю поэкспериментировать с сочетанием различных гарнитур и семейств шрифтов, но для получения хорошего результата надо обладать глубокими знаниями и опытом. Особенно если вы собираетесь применять шрифты из разных семейств. Если сомневаетесь, не усложняйте.

> **СОВЕТ**
>
> На странице www.mezzoblue.com/archives/2004/01/18/type_the_ext есть неплохая статья о применении этой техники и смежных вопросах.

Управление сглаживанием

Внимательно приглядитесь к графическим шрифтам, обратите внимание на то, как четко очерчены литеры. Возможно, вы ранее задавались вопросом, почему некоторые графические шрифты выглядят в Web слегка смазанными, – тому есть несколько причин. Во-первых, с самого начала мог быть неверно выбран формат графического файла и плохо проведена его оптимизация. Другая же причина состоит в сглаживании (anti-aliasing).

Сглаживанием называется процедура смягчения состоящих из пикселей изогнутых штрихов литер шрифта путем добавления полутонов на краях. В типографских программах имеются замечательные средства, позволяющие получить прекрасно выглядящий конечный продукт, но дизайнеры, желающие, чтобы текст смотрелся более четко, могут копнуть поглубже и самостоятельно управлять сглаживанием.

Чем меньше размер шрифта, тем это становится важнее. Штрихи, из которых состоит литера (а в таких шрифтах, как Morpheus, штрихов много) нельзя сделать меньше одного пикселя. Процедура сглаживания добавляет полутоновые пиксели, из-за чего возникает нежелательное размывание. Но можно воспользоваться имеющейся в большинстве серьезных графических программ функцией *kern* (кернинг), которая немного увеличивает расстояние между литерами, чтобы результат сглаживания выглядел получше. Подробнее о сглаживании и кернинге можно прочитать на странице www.wpdfd.com/wpdtypo3.htm.

Углубимся в типографию

Если все эти разговоры о типографии вызвали у вас интерес, что ж – есть много литературы на эту тему, как в печатном, так и электронном виде.

Книги

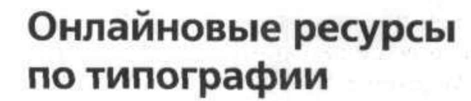
В следующий раз, когда пойдете в книжный магазин, задержитесь у полки, посвященной дизайну. Там вы найдете немало книг, касающихся типографии. Некоторые из них повествуют об истории вопроса, другие – о дизайне шрифтов, в третьих иллюстрируются работы конкретного дизайнера. Просто пролистайте понравившиеся книги – так же, как вы просматриваете исходные тексты на языках HTML и CSS, это прекрасный способ познакомиться с методами и идеями,

о которых вы раньше ничего не знали. Не пропустите следующие книги хорошо известного автора Робина Уильямса (Robin Williams):

- *The Mac Is Not a Typewriter, 2nd Edition* и *The PC Is Not a Typewriter* (обе – Peachpit Press, 2003). Первая книга посвящена компьютерам Macintosh, вторая – ПК под управлением Windows. Обе содержат прекрасное изложение базовых принципов использования шрифтов на экране и в печатных материалах.
- *The Non-Designer's Type Book* (Peachpit Press, 1998). Вне зависимости от того, являетесь ли вы профессиональным дизайнером или только собираетесь им стать, эта книга откроет вам некоторые секреты, касающиеся шрифтов, подскажет, как правильно оформлять документы и когда имеет смысл нарушить эти правила.

Более опытным дизайнерам будут интересны и другие книги о типографии. Вот несколько заслуживающих внимания названий:

- *Typographic Design: Form and Communication, 3rd Edition.* Rob Carter, Ben Day и Philip Meggs (John Wiley & Sons, 2002). Содержит подробную историю развития типографского дела, изучение литер и справочник наиболее широко распространенных шрифтов. Книга не рассчитана на начинающих, но послужит прекрасным подспорьем для желающих углубить свои знания.
- *The Elements of Typographic Style, 2nd Edition.* Robert Bringhurst (Hartley & Marks Publishers, 2002). Полный обзор типографского дела. Материал представлен с энциклопедической глубиной, предназначен для профессиональных типографов, но главы о практических применениях теории делают книгу поистине бесценной.
- *Alphabet: The History, Evolution, and Design of Letters We Use Today.* Allan Haley (Pub Overstock Unlimited, 1995). Написанная автором многочисленных трудов по типографии, эта книга увлекательно рассказывает об истории и дизайне современных английских шрифтов. Рекомендуем и другие книги Хэйли, они рассчитаны на любой уровень подготовки.
- Поинтересуйтесь и современными работами, вызывающими споры, например книгой *The End of Print: The Grafik Design of David Carson.* Revised Edition, Lewis Blackwell и David Carson (Chronicle Books, 2000). В ней вы найдете неиссякаемый источник вдохновения и аргументов.



Онлайновые ресурсы по типографии

В Web есть множество ресурсов, посвященных типографии, начиная от архивов шрифтов до руководств и интервью. Задайте поисковой системе Google запрос *typography, type foundries,* и вы получите кучу ссылок, дня не хватит просмотреть их. Приведем несколько избранных ресурсов:

- Type Designers (http://cgm.cs.mcgill.ca/~luc/longdesigners.html). Алфавитный указатель, составленный и поддерживаемый Люком Девроем (Luc Devroye). Содержит сведения о дизайнерах шрифтов от A до Z с описанием их работ и ссылками на сайты.
- Identifont (www.identifont.com). Очень полезный сайт, на котором вы сможете найти нужный шрифт. Там же предлагаются и другие типографские ресурсы.
- MyFonts (www.myfonts.com). Возможно, самый плохо выглядящий сайт в Web, но полезной информации на нем море. Поисковая система позволяет проводить свободный поиск по словам, а инструмент WhatTheFont пытается определить, какой шрифт использован в представленном вами изображении.
- Adobe Type Library (http://store.adobe.com/type). На сайте компании Adobe имеется первоклассная библиотека коммерческих шрифтов.
- Fonts, Main на сайте Website Tips (www.websitetips.com/fonts). Каталог всего, имеющего отношение к шрифтам для Web. Предлагаются бесплатные и условно бесплатные шрифты, иноязычные шрифты, перечни конференций, посвященных вопросам типографии, а также программы, связанные со шрифтами.

Учиться типографскому делу можно всю жизнь, но для отчаяния нет причин. Чем больше вы знаете, тем лучших результатов сможете добиться в своих работах.

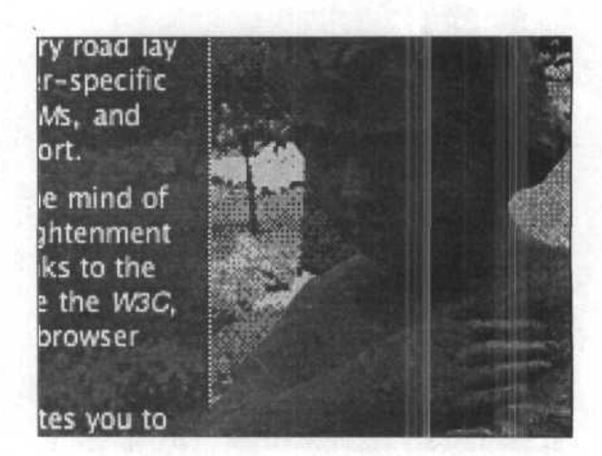
Главное – это баланс

Аристотель учил, что мудрец соблюдает умеренность во всем. Хотя временами дизайн и должен быть экстремальным, но большинство дизайнеров-профессионалов стараются соблюдать баланс.

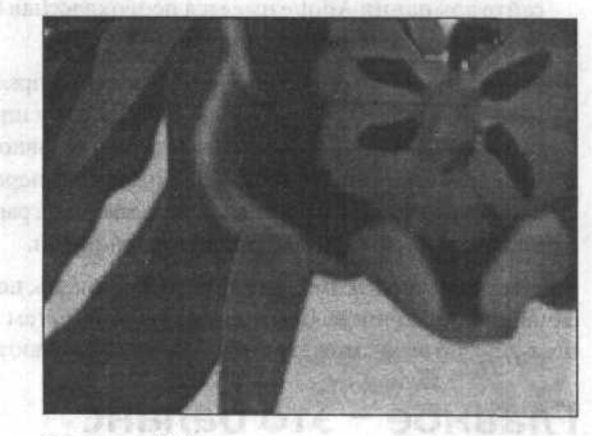
Контрастирующие гарнитуры в «Золотой середине» ясно показывают, что, отыскав удачную комбинацию шрифтов, веб-дизайнер сможет достичь желаемого равновесия, золотой середины между практической целью и надолго запоминающимся, утонченным зрительным образом.



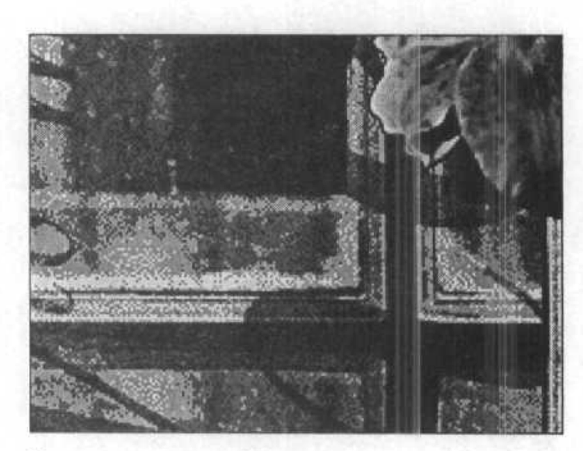
Кукурузные хлопья, с. 216



Удвоение, с. 222



Тюльпан, с. 234

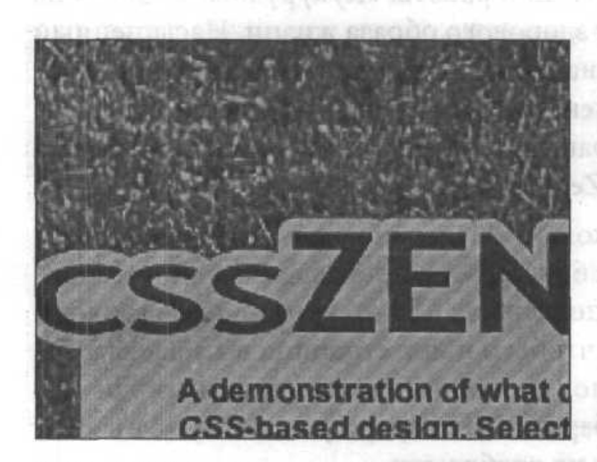


Калитка в мой сад, с. 240

6

Специальные эффекты

Небо над бонсай, с. 228



Упругая лужайка, с. 246

Из этой главы вы узнаете, как создавать различные эффекты с помощью фоновых изображений, прозрачности и организации слоев, чтобы добиться того, что раньше в Web считалось недостижимым.

Быть может, самое важное, что вы найдете в этой главе, – это информация о больной для каждого дизайнера теме: поддержке в разных браузерах. На примере рассматриваемых здесь работ вы узнаете о различных приемах, обходных путях и иных способах справиться с конкретным браузером.



Дизайнер **Шон Инман** (Shaun Inman)
www.csszengarden.com/057

Кукурузные хлопья

Каскад и эффект наложения слоев

Шон Инман очень любит сбалансированный завтрак. Вооружившись коробкой кукурузных хлопьев, капелькой юмора и твердым желанием нарушить правила, Инман положил образы, навеянные утренней едой, в основу прелестной работы «Кукурузные хлопья» на тему здорового образа жизни. Насыщенный специальными эффектами, видимыми в современных браузерах, этот дизайн раздвигает границы, определяемые разметкой сайта CSS Zen Garden.

Поскольку очень многие до сих пор пользуются браузерами, не обеспечивающими полной поддержки CSS, то Инман позаботился о том, чтобы и в них страница выглядела прилично, но особо впечатляющие эффекты приберег для тех браузеров, которые в состоянии их отобразить.

То, что в большинстве веб-сайтов показалось бы спорным, для Zen Garden подходит как нельзя лучше, поскольку этот сайт всецело посвящен экспериментированию во благо прогресса.

СОВЕТ

Подробнее о специфичности и наследовании можно прочитать в спецификации CSS, конкретно в разделе «Assigning property values, Cascading and Inheritance»(www.w3.org/TR/REC-CSS2/cascade.html).

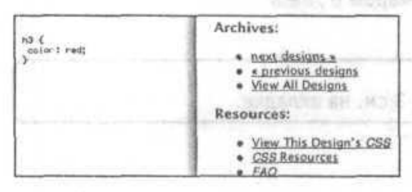


Рис. 1 ✦ Все элементы h3 на странице отображаются красным цветом

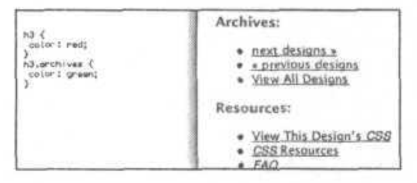


Рис. 2 ✦ Элементы h3 класса archives отображаются зеленым цветом, все остальные – по-прежнему красным

ПРИМЕЧАНИЕ

Цветной вариант рис. 1 и 2 см. на вкладке.

Каскад

Меню и эффекты прозрачности, используемые в «Кукурузных хлопьях», видны только в браузерах с высоким уровнем поддержки спецификации CSS2, например в Firefox. Для выборочного применения этих возможностей необходимы CSS-фильтры, основанные, в частности, на тонкостях механизма каскадирования.

Что же понимается под каскадированием в каскадных таблицах стилей? Согласно W3C, «с помощью механизма каскадирования в CSS каждому правилу назначается определенный вес. Если применимо несколько правил, то приоритет отдается тому, у которого вес наибольший». Две взаимосвязанные концепции – специфичность и наследование – могут стать как вашими лучшими друзьями, так и самыми страшными кошмарами... не обязательно именно в этом порядке. CSS позволяет применять правила сразу к нескольким элементам. Не будь такого механизма, в таблицах стилей пришлось бы устанавливать взаимно-однозначное соответствие между правилами и элементами – это непрактично и громоздко.

Наследование

Существует возможность применить стиль к обобщенным элементам и наборам элементов, удовлетворяющих некоторому условию. Так, правило, заданное для элементов h3, будет применяться ко *всем* элементам h3 на странице (рис. 1):

```
h3 {
  color: red;
}
```

Таким образом, все элементы h3 на странице будут отображаться красным цветом. Все просто. Но что, если надо отменить это правило для некоторых элементов? Придется ввести дополнительный класс (рис. 2):

```
h3 {
  color: red;
}
h3.archives {
  color: green;
}
```

Первое стилистическое правило окрашивает все элементы h3 в красный цвет, второе переопределяет его только для тех элементов, для которых указан класс archives.

Пока ничего особо сложного нет, но сделаем еще один шаг. При наличии следующей таблицы как будут выглядеть элементы h3 в разделе #linkList (рис. 3)?

```
h3 {
  font: 1em Verdana, sans-serif;
  color: red;
}
h3.archives {
  color: green;
}
#linkList h3 {
  font-size: 0.8em;
}
```

Они отображаются красным цветом и шрифтом Verdana размером 0,8em (или стандартным шрифтом без засечек, если Verdana отсутствует). Поскольку семейство шрифтов не переопределено в последнем правиле, то Verdana наследуется от определенного выше глобального правила для элементов h3; изменяется только размер. А в предположении, что рассматриваемый элемент h3 имеет класс archives, он будет окрашен в зеленый цвет.

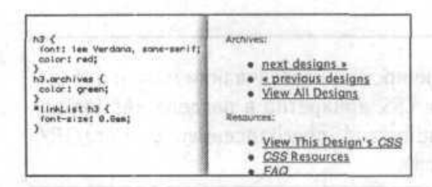


Рис. 3 ✦ Сочетание стилей приводит к тому, что заголовок отображается зеленым цветом и шрифтом Verdana размером 0,8em

ПРИМЕЧАНИЕ

Цветной вариант рис. 3 см. на вкладке.

Специфичность

В предыдущем примере продемонстрирована также концепция специфичности – последнему правилу назначен больший приоритет, чем первому. Поскольку в нем задан еще и класс, то правило считается более специфичным и «весит» больше.

Если бы понятия специфичности не было, то возникла бы конкуренция между двумя правилами, ведь для элементов h3 в одном правиле определен размер 0,8em, а в другом – 1em. Специфичность – это простой и надежный способ однозначно выявить победителя.

На первый взгляд, ничего сложного, но со специфичностью связаны некоторые проблемы. Вполне возможно создать чрезмерно специфичное правило, отменить которое можно будет, только задав еще более специфичное.

Рассмотрим пример:

```
body>html #header ul.navigation li.home a {
  color: red;
}
```

Что произойдет, если вы захотите перекрасить ссылки, находящиеся в состоянии :hover, в синий цвет? Придется пойти еще дальше: продублировать и без того

длинный список селекторов и добавить к нему дополнительный селектор, выделяющий состояние :hover:

```
body>html #header ul.navigation li.home
a:hover {
 color: red;
}
```

Специфичность под контролем

Если все спланировать заранее, то удастся избежать сложных, чрезмерно специфичных селекторов. Составляя стилистические правила, стремитесь к максимальной общности, а специфику добавляйте, лишь когда в этом появляется реальная необходимость.

Сравните, например, селектор

```
#linkList ul li h3 {
 color: red;
}
с таким:
#linkList h3 {
 color: red;
}
```

В первом примере указано много критериев, которым должен удовлетворять элемент h3, рисуемый красным цветом: он должен находиться внутри неупорядоченного списка и являться отдаленным потомком элемента с конкретным идентификатором. Во втором примере условие нахождения внутри списка снято, но требование иметь в качестве предка элемент с указанным идентификатором осталось. Если начать с достаточно общих условий и постепенно продвигаться к более специфичным, то не возникнет необходимости составлять слишком длинные и чересчур специфичные правила.

Поиск ошибок, связанных со специфичностью

Вот простой совет по поводу того, как искать ошибки в верстке на базе CSS: убедитесь, что причина не кроется в конфликте между правилами с разной специфичностью. Если применение некоторого правила не дает желаемого эффекта, первым делом проверьте, достаточно ли специфично это правило. Для этого часто бывает полезно вставить избыточную временную разметку и посмотреть, выбирает ли селектор рассматриваемый элемент. Рассмотрим пример:

```
#linkList h3 {
 border: solid 20px red;
}
```

Если у элемента h3 не оказывается рамки, сделайте селектор более специфичным. Если же рамка появляется, то уберите объявление border из правила; ошибка, скорее всего, не связана со специфичностью.

Переопределение специфичности

В CSS встроена возможность переопределить специфичность для любого стилистического правила. Объявление !important, применяемое на уровне отдельных атрибутов, гарантирует, что данное правило будет применено безусловно, какова бы ни была специфичность других селекторов в таблице стилей. Например:

```
h3 {
 color: red !important;
}
#linkList h3 {
 color: blue;
}
```

Хотя второе правило более специфично для элементов h3, находящихся внутри раздела #linkList, они все равно будут отображаться красным цветом, так как объявление !important в первом правиле заставляет игнорировать уровень специфичности. Отметим, впрочем, что Microsoft Internet Explorer для Windows не поддерживает конструкцию !important, так что полагаться на это поведение в общем случае не стоит.

Однако отсутствие поддержки в Internet Explorer можно считать преимуществом при отладке CSS; использование объявления !important с целью применения некоторых правил во всех браузерах, кроме Internet Explorer, – это одна из возможностей фильтрации. В примере выше рассматриваемые элементы h3 будут отображаться красным в таких браузерах, как Firefox, Opera и Safari, но синим в Internet Explorer.

Каскадирование активно используется в «Кукурузных хлопьях», чтобы компенсировать отсутствие определенных эффектов в некоторых браузерах. Ко всему прочему, Инман так отфильтровал стили, что в браузерах, которым недостает необходимой поддержки, и только в них, появляется нахальное сообщение (рис. 4), предлагающее загрузить современный браузер и насмехающееся над Internet Explorer.

Слои

Сообщение, видное только в Internet Explorer, и прочие элементы страницы позиционированы абсолютно. Чтобы избежать перекрытия, пришлось предпринять некоторые меры.

Рис. 4 ✦ Это сообщение появляется у пользователей, которые просматривают «Кукурузные хлопья» в браузере Internet Explorer

Для управления порядком размещения вдоль оси z в CSS применяется атрибут `z-index`. Так как абсолютно позиционированные элементы исключаются из нормального потока документа, то стандартного метода управления перекрытием часто оказывается недостаточно. Не будь свойства `z-index`, позиционированные элементы выстраивались бы снизу вверх: тот элемент, который следует в HTML-разметке первым, оказался бы внизу стека, а последующие элементы – поверх него.

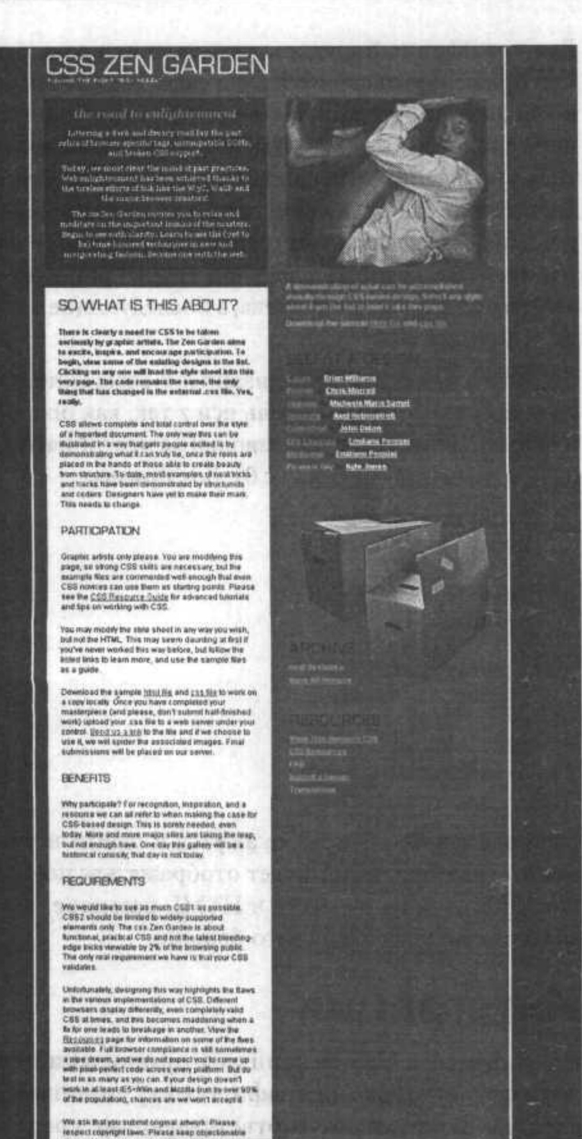
Свойство `z-index` дает дизайнеру возможность управлять размещением вдоль оси z так, как он считает нужным. Допустимы произвольные неотрицательные целые значения (начиная с 0). Вот пример:

```
#linkList h3 {
 position: absolute;
 top: 50px;
 left: 50px;
 z-index: 10;
}
#linkList ul {
 position: absolute;
 top: 50px;
 left: 50px;
 z-index: 20;
}
```

Если элементы `ul` и `h3` окажутся в одном и том же месте страницы, то `ul` будет отображаться поверх `h3` вне зависимости от места в HTML-разметке, так как значение `z-index` для него больше.

Интеграция

Имеющиеся в CSS механизмы организации слоев должны быть знакомы графическому дизайнеру, поскольку напоминают соответствующие средства в программах манипулирования изображениями. Каскадирование – это, скорее, концепция из языков программирования, не имеющая прямых аналогий в мире дизайна, но ее полезно освоить, поскольку это позволяет точно управлять взаимодействием различных элементов, что и демонстрирует работа «Кукурузные хлопья».



Дизайнер **Егор Клоос** (Egor Kloos)

www.csszengarden.com/062

Удвоение

Два дизайна – одна таблица стилей

Отсутствие ограничений на сайте CSS Zen Garden навело Егора Клооса на мысль продемонстрировать, что CSS – это инструмент для воплощения в жизнь любых мыслимых идей. Заинтересовавшись трюками, позволяющими отправлять таблицы стилей с нетривиальными конструкциями только тем браузерам, которые могут их поддержать, Клоос убедительно доказал, что изобретательный код может привести к поразительным результатам. Так и появилась на свет работа «Удвоение», в которой единственная таблица стилей порождает два совершенно разных дизайна.

ПРИМЕЧАНИЕ

Аббревиатура MOSe означает Mozilla Opera Safari enhancement (дополнения в браузерах Mozilla Opera Safari). Подробнее см. на странице www.mezzoblue.com/archives/2003/06/25/.

ПРИМЕЧАНИЕ

Селектор дочерних элементов позволяет выбрать только непосредственных потомков данного элемента, а не потомков с любым уровнем вложенности.

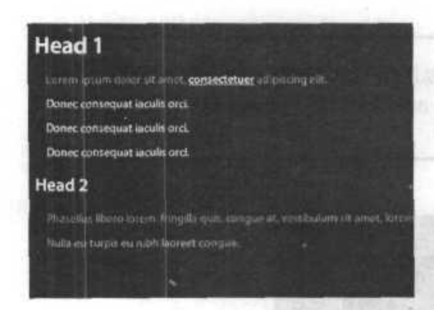


Рис. 1 ✦ Обратите внимание, что селектор дочерних элементов выбирает лишь непосредственных потомков родителя, а не потомков с более глубокой вложенностью

MOSe

Споры по поводу того, как улучшить дизайн при просмотре в браузерах с более полной поддержкой CSS, велись еще до появления «Удвоения». Идея проста: воспользоваться отсутствием поддержки некоторых селекторов в том или ином браузере для того, чтобы именно с их помощью составить правила, которые увидят браузеры, имеющие необходимую поддержку, а точнее Mozilla и его варианты Firefox, Opera и Safari.

Конкретно, техника MOSe основана на селекторах, которые не поддерживаются в Microsoft Internet Explorer. Наиболее часто для этой цели применяются *селекторы дочерних элементов*, *селекторы сестринских элементов* и *селекторы атрибутов*.

Селекторы дочерних элементов

Селекторы дочерних элементов, как следует из названия, позволяют выбрать непосредственных потомков данного элемента. Она похожи на селекторы потомков с тем, однако, отличием, что селекторы потомков выбирают потомков с произвольным уровнем вложенности, а не только непосредственных.

Синтаксически в селекторе дочерних элементов сначала указывается родительский элемент, затем следует комбинатор > (*комбинатором* называется символ, служащий для объединения частей селектора), за ним – дочерний элемент и, наконец, само объявление:

```
div#content>p {color: orange;}
```

Это правило говорит, что все абзацы, являющиеся непосредственными потомками `div` с идентификатором `#content`, будут выкрашены в оранжевый цвет, (на рис. 1 они показаны слегка затемненным шрифтом).

Селекторы сестринских элементов

Селектор сестринских элементов выбирает сестер (или, если хотите, братьев) данного элемента (то есть имеющих того же родителя, что и этот элемент).

Синтаксически сначала указывается начальный элемент, затем комбинатор +, за ним – сестринский элемент и, наконец, объявление:

```
div#warning p + p {color: red;}
```

В результате абзацы, соседствующие с начальным абзацем внутри div с идентификатором #warning, будут отображаться красным цветом. Если применить это правило к той же разметке, что и выше, то мы увидим картину, изображенную на рис. 2 (красный шрифт отображен серым цветом).

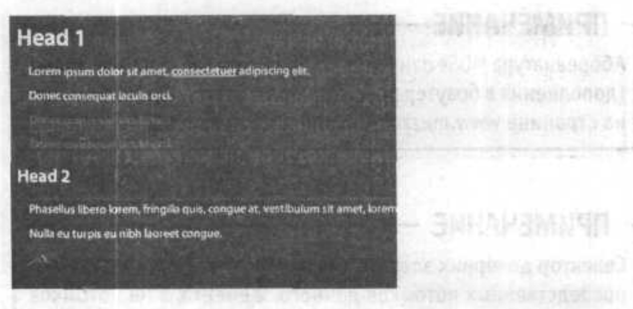


Рис. 2 ✦ Селекторы сестринских элементов выбирают элементы с тем же родителем, что и у начального

> **ПРИМЕЧАНИЕ**
> На сайте группы OPAL (http://gallery.theopalgroup.com/selectoracle) имеется прекрасное объяснение работы селекторов в CSS.

Селекторы атрибутов и сопоставление с образцом

А вот тут-то и начинается самое интересное! Возможно, читатель незнаком с селекторами атрибутов, и прежде всего потому, что они не поддерживаются в Internet Explorer. Поэтому большинство практикующих веб-дизайнеров даже не думают о том, как их можно использовать. А ведь селекторы атрибутов предлагают очень интересные возможности.

Селектор атрибутов позволяет выбирать элементы в зависимости от того, какие атрибуты для них заданы и, что еще более полезно, от того, каковы значения этих атрибутов. Так, если имеется ссылка с конкретным URL, то можно сделать выбор, базируясь не только на том факте, что имеется атрибут href, но и на его значении. Тогда к каждому элементу с данным URL будет применен соответствующий стиль.

Рассмотрим следующую таблицу стилей:

```
[id] {color: teal;}
[id="warning"] Pcolor: red;}
[href~="http://www.molly.com/"] {text-
decoration: none;}
```



Рис. 3 ✦ Действие селекторов атрибутов в совместимом браузере

> **ПРИМЕЧАНИЕ**
> Цветной вариант рис. 3 см. на вкладке.

На рис. 3 показано, как эти селекторы модифицируют внешний вид элементов в зависимости от атрибутов и их значений.

Именно этим видом селекторов и воспользовался Клоос, чтобы дизайн в совместимых с CSS браузерах был одним, а в Internet Explorer – совершенно другим.

CSS-сигнатуры

CSS-сигнатуры получают все более широкое распространение. Идея в том, чтобы приписать идентификатор элементу body, который станет затем уникальной сигнатурой для различных таблиц стилей.

ПРИМЕЧАНИЕ

Применение техники MOSe может основываться и на других селекторах, не поддерживаемых Internet Explorer, в том числе и на некоторых конструкциях из спецификации CSS 3.0. Полный обзор способов применения MOSe для улучшения дизайна см. на странице www.mezzoblue.com/archives/2003/06/25/ mose.



Рис. 4 ✦ Специальное поведение ссылок в «Удвоении»

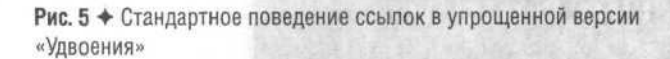


Рис. 5 ✦ Стандартное поведение ссылок в упрощенной версии «Удвоения»

Истинная ценность CSS-сигнатуры в том, что она дает опытному пользователю возможность перенастроить внешний вид чужого сайта. Например, если вы часто заходите на сайт, в котором текст отображается шрифтом размером 9 пикселей и светло-серым цветом на белом фоне, так что вы его читать не в состоянии, то вовсе необязательно забрасывать автора сайта письмами типа «исправь, а не то...». При условии, что автор включил CSS-сигнатуру, вы сможете переопределить размер шрифта с помощью пользовательской таблицы стилей в своем браузере.

Увидеть, как CSS-сигнатура применяется в Zen Garden, можно на примере работы Клооса:

```
<body id="css-zrn-garden">
```

Чтобы создать дизайн «Удвоения» с дополнительными возможностями, Клоос воспользовался сигнатурой элемента `body`. Заглянув в таблицу стилей, вы увидите необычные, на первый взгляд, конструкции:

```
body[id=css-zen-garden] a:link { color: #f90; }
body[id=css-zen-garden] a:visited { color: #f90; }
body[id=css-zen-garden] a:hover,
body[id=css-zen-garden] a:active { color: #fff; }
```

Здесь Клоос определил поведение ссылок в предположении, что они находятся внутри элемента `body`, для которого задан атрибут `id` с указанным значением (рис. 4).

Сравнение MOSe со стилями для Internet Explorer

Предложить браузеру Internet Explorer совершенно другой набор стилистических правил нетрудно. Нужно лишь позаботиться о том, чтобы он мог их интерпретировать.

Вот как выглядит версия таблица стилей для Internet Explorer:

```
a:link { color: #f90; }
a:visited { color: #f90; }
a:hover, a:active { color: #fff; }
```

Нетрудно заметить, что сами правила не отличаются от приведенных выше, но такой синтаксис Internet Explorer поддерживает, поэтому они годятся для упрощенной версии дизайна (рис. 5).

8 Зак. 1942

Версия «Удвоения» с дополнениями

В версии дизайна с дополнениями есть несколько заслуживающих упоминания эффектов. Построен он по аналогии со слоями в Adobe Photoshop: фон замощен повторяющимся орнаментом, а прочие элементы расположены поверх него.

Фон задается для элемента body (рис. 6):

```
body[id=css-zen-garden]
{ margin: 100px 0 0 0; padding: 0; text-align:
center; background:
transparent url(squidback.gif); }
```

Далее добавляется еще одно изображение, на этот раз оно ассоциируется с элементом #container. Картинка выступает в роли задника для области контента (рис. 7).

В преамбуле интересный эффект создается за счет использования светло-серого фона, через который просвечивает задник (рис. 8):

```
body[id=css-zen-garden] #preamble        {
position:
absolute; top: 30px; left: 20px; display:
block; margin: 0; border:
1px dotted #fff; paddingL 0; width: 196px;
height: 290px; background:
transparent url(blk35.png) repeat; overflow:
hidden; }
```

Такой же эффект используется и в области контента, где поверх фоновой картинки отображается полоса прокрутки. Наконец, для расположенной справа панели навигации со вкладками применен селектор :hover, создающий очень красивый эффект выделения (рис. 9). Конечно, это возможно лишь в браузерах, поддерживающих :hover для любых элементов, а не только для ссылок, но можно считать, что браузеры, поддерживающие селекторы, на которые опирается MOSe, этому условию удовлетворяют.

Поскольку вниманию Internet Explorer предлагается куда более традиционная таблица стилей, то и говорить об эффектах не приходится. Самый интересный аспект дизайна для Internet Explorer – это заключенный в нем юмор. В нем скрыт намек на модель прямоугольных областей в CSS – девушка с удивленным выражением на лице, заключенная в ящик, – и добродушная насмешка над ошибкой в интерпретации этой модели в Internet Explorer 5.x (рис. 10). (Модель прямоугольных областей – это визуальная модель, которую браузеры применяют для вычисления положения отдельных элементов на странице. Подробнее см. www.w3.org/TR/REC-CSS2/box.html).

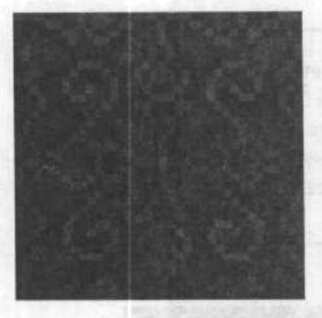


Рис. 6 ✦ Этот фрагмент, взятый с сайта squidfingers.com, образует фон



Рис. 7 ✦ Эта картинка выступает в роли задника для области контента в дополненной версии дизайна

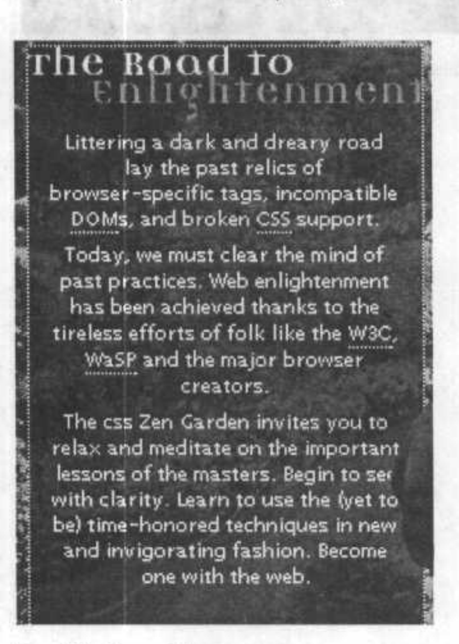


Рис. 8 ✦ Этот эффект создан за счет двух графических слоев, из которых верхний прозрачен



Рис. 9 ✦ Применение селектора динамического псевдокласса `:hover` к элементам позволяет добиться эффекта выделения без привлечения сценариев



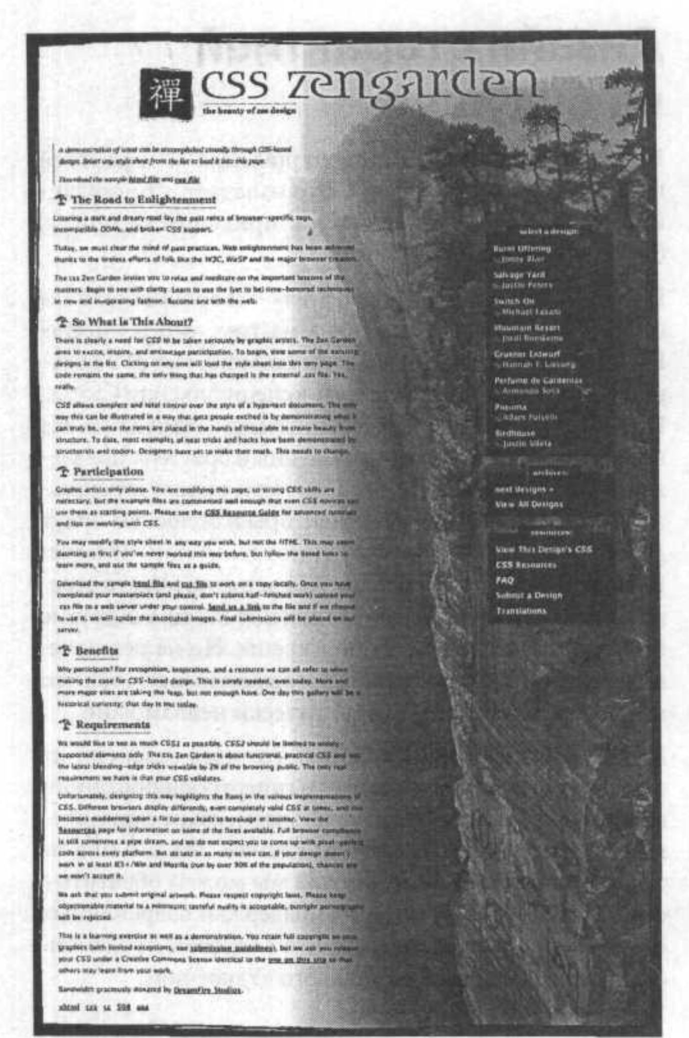
Рис. 10 ✦ «Модель прямоугольных областей»

Дизайн с гарантией на будущее

Не затихают споры по поводу дизайна «с гарантией на будущее» (future-proof). Это означает, что сделанное сегодня будет работать и с приходом будущих, более передовых технологий.

Понятно, что вся идея «Удвоения» – и вообще техника MOSe как таковая – основана на том, что в Internet Explorer плохо поддерживаются ставшие уже стандартными селекторы, определенные в спецификации CSS 2.1. Но что произойдет в туманном будущем, когда Microsoft все-таки выпустит более современный браузер с поддержкой всех этих средств? Теоретически ничего страшного – при условии, что все селекторы и эффекты будут реализованы, как должно, гипотетический Internet Explorer для Windows версии 7 будет отображать страницу так, как это уже сейчас делают Mozilla и другие браузеры. И тогда мир станет краше. Но не исключено, что возникнет очередная несовместимость, так что предсказать результаты практически невозможно.

Клоос придерживается мнения, что идея гарантии на будущее – это «продукт чьего-то гипертрофированного воображения», поскольку мы не знаем, какие сюрпризы может преподнести это самое будущее. Так это или не так, но, по крайней мере, сегодня мы можем обратить в свою пользу факт отсутствия поддержки современных селекторов в Internet Explorer, что с блеском продемонстрировано на примере дуального «Удвоения».



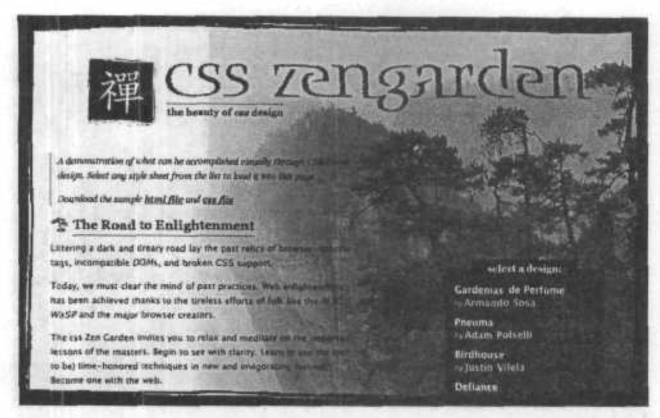
Дизайнер **Майк Дэвидсон** (Mike Davidson)
www.csszengarden.com/069

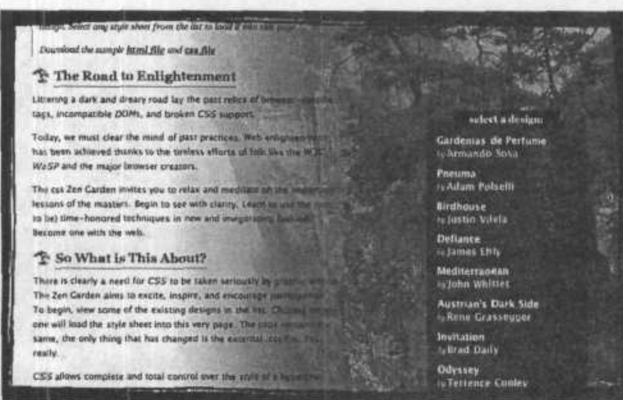
Небо над бонсай

Преодоление ограничений браузеров с помощью изобретательного подбора графики и кода

Поднимая тему озарения на новую высоту, Майк Дэвидсон создал обитель дзен-буддизма на небесах. Взяв за основу фотографию утеса, поросшего деревцами, он для начала прикинул, как добиться баланса текста с доминирующим изображением.

Поместив изображение утеса в правый верхний угол и осветлив его, Дэвидсон расположил текст слева и тем самым избежал проблем контрастностью, которые могли бы возникнуть, если бы текст шел поверх картинки. Черный текст с широкими пропусками занимает пустое белое пространство слева от утеса, и взгляд зрителя плавно скользит вниз, где его ожидает сюрприз: стилизованные логотипы основных браузеров, выгравированные на поверхности утеса, и скалолаз, заглядывающий в пропасть. Скалолаз – это сам Дэвидсон. Успешно одолев подъем к вершинам CSS, он понимает, что пути назад нет.





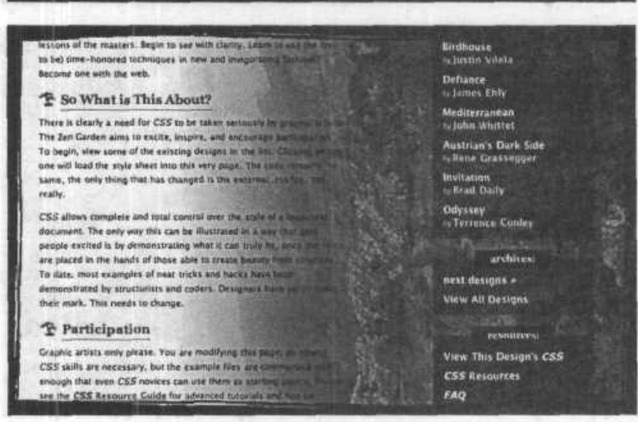


Рис. 1 ✦ Рамка остается неподвижной во время прокрутки страницы

Управление версиями

Некоторые из наиболее интересных графических эффектов в «Небе над бонсай» видны не во всех браузерах – вследствие использования Дэвидсоном специальных свойств CSS оказалось невозможно представить одну и ту же картину всем зрителям. Вместо того чтобы отказываться от части визуальных эффектов, Дэвидсон обманул браузеры, которые показывали их неправильно, воспользовавшись CSS-фильтрами.

Положение черной рамки

«Небо над бонсай» окаймлено толстой черной рамкой с фиксированным позиционированием:

```
#extraDiv2 {
  background-image: url(edge_bottom_black.gif);
  background-repeat: repeat-x;
  position: fixed;
  bottom: 0;
  left: 100%;
  width: 100%;
  height: 20px;
  margin-left: -100%;
 }
```

Атрибут позиционирования `fixed` позволяет точно расположить элемент на странице, так что он будет оставаться на месте даже при ее прокрутке. Создается впечатление, что такие элементы всплывают поверх остальной части страницы, именно так ведет себя рамка в рассматриваемом дизайне (рис. 1).

Однако фиксированное позиционирование нельзя считать надежной техникой, поскольку некоторые браузеры (и, прежде всего, Internet Explorer для Windows) прокручивают фиксированные элементы вместе с остальной страницей. Иногда такое ухудшение замысла можно считать приемлемым: если браузер оставляет элементы с атрибутом `fixed` на месте – прекрасно, если нет – пусть прокручиваются, потерпим.

Обходной маневр для позиционирования черной рамки

Но в случае «Неба над бонсай» рамка наползла бы на контент, если бы фиксированные элементы прокручивались (рис. 2), поэтому необходимо было найти какой-то выход из этого положения. Существуют различные фильтры, позволяющие скрыть часть таблицы стилей от некоторых браузеров, но вместо того, чтобы полностью убирать рамку, Дэвидсон придумал, как убрать ее за пределы экрана, воспользовавшись совсем другим дефектом браузеров.

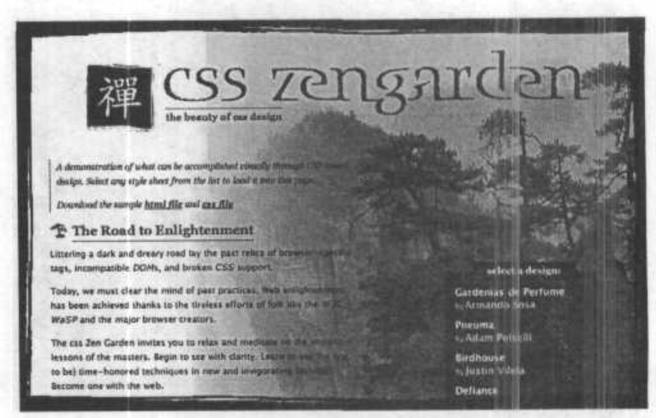
Каждая из двух горизонтальных частей рамки – верхняя и нижняя – позиционируется с атрибутом `fixed` и смещением влево 100%. Поэтому начальные позиции расположены как раз за правой границей окна браузера, то есть не видны. Чтобы они все-таки появились на экране, Дэвидсон задал для каждой части рамки значение левого поля `margin-left` –100%.

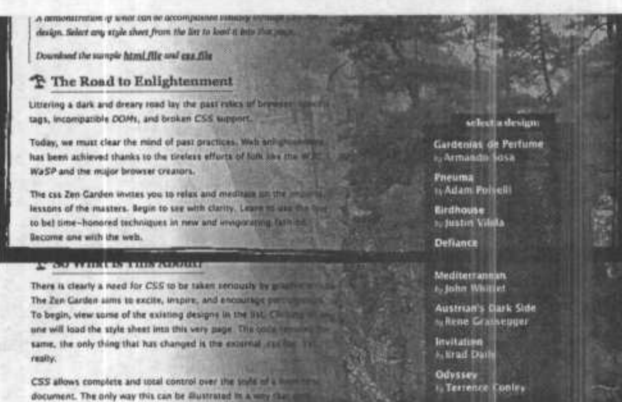
```
#extraDiv2 {
  background-image: url(edge_bottom_black.gif);
  background-repeat: repeat-x;
  position: fixed;
  bottom: 0;
  left: 100%;
  width: 100%;
  height: 20px;
  margin-left: -100%;
}
```

Поскольку большинство современных браузеров поддерживает отрицательные значения полей, то рамка оказывается именно там, где и хотел Дэвидсон. Все прочие браузеры не вернут рамку на экран, так что она останется скрытой. К счастью для «Неба над бонсай», те же самые браузеры не поддерживают и фиксированного позиционирования.

Будда

Решив вопрос с размещением рамки, Дэвидсон добавил еще один визуальный эффект, чтобы усилить звучание основной темы. На выступающем камне, рядом с правым краем окна браузера находится золотая статуэтка сидящего Будды. Это изображение было позиционировано с атрибутом `fixed` с применением той же техники, что и для рамки.





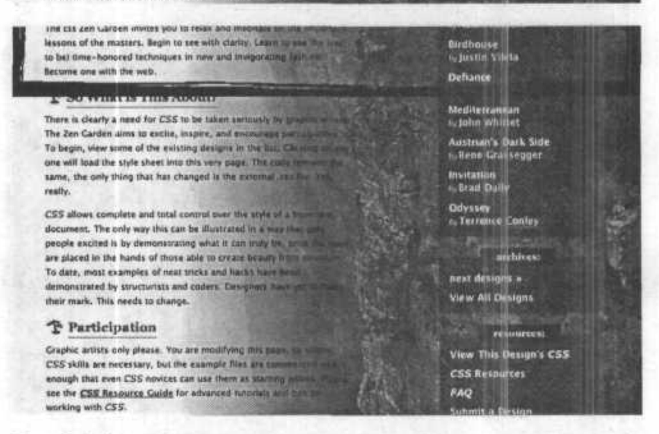


Рис. 2 ✦ Рамка наползает на контент, если ее разрешено прокручивать

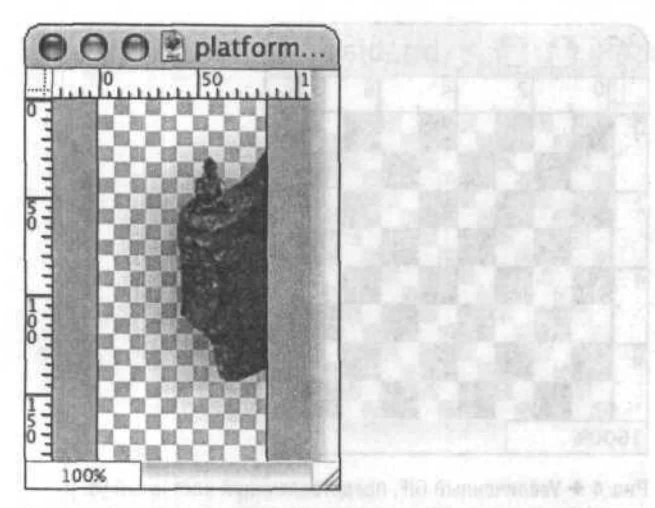


Рис. 3 ✦ За уступом и статуэткой Будды, благодаря прозрачности в формате PNG, видна слабая тень

Поскольку Internet Explorer в любом случае не смог бы правильно показать статуэтку, Дэвидсон решил воспользоваться альфа-прозрачностью, предусмотренной форматом PNG, которая в Internet Explorer также не поддержана.

За уступом в PNG-файле была помещена слабая прозрачная тень (рис. 3). Когда пользователь прокручивает «Небо над бонсай», тень не дает уступу затеряться среди камней на заднем фоне, так что создается впечатление трехмерности.

Этот простой, но тонкий штришок невозможно было бы воспроизвести с применением ограниченной прозрачности GIF-файла, и, следовательно, он не виден в браузерах, не умеющих правильно обрабатывать прозрачность в формате PNG.

Хотя парящий уступ и изображение статуэтки усиливают зрительное впечатление от «Неба над бонсай», но Дэвидсон так глубоко прочувствовал тему озарения, что захотел, чтобы лишь истинно «просветленные» браузеры с полной поддержкой CSS могли ее правильно показать. Воспользовавшись тем же трюком с отрицательными полями, что и для рамки, Дэвидсон просто скрыл весь парящий уступ от Internet Explorer для Windows.

Однако понимая, что многие посетители, работающие именно с этим браузером, не смогут наблюдать эффект, он сделал так, что остальная часть дизайна выглядит пристойно даже без уступа. В данном случае некоторая деградация приемлема, хотя трудно отрицать, что работа смотрится гораздо лучше в браузерах, поддерживающих фиксированное позиционирование и PNG-прозрачность.

Прозрачность бокового блока

Разместив текст и фотографию утеса, Дэвидсон столкнулся с проблемой: нужно было как-то расположить списки ссылок: select a design, archives и resources. Потратив много времени на шлифовку изображения утеса, он не хотел чем-то закрывать его.

Перекрытие с небольшим уровнем прозрачности казалось хорошим решением, но PNG-прозрачность здесь не годилась: на этот раз было важно, чтобы и пользователи Internet Explorer для Windows могли видеть всю картинку. Поэтому он обратился к старому

трюку, который веб-дизайнеры давно применяли для создания иллюзии полупрозрачности с помощью небольшого прозрачного GIF-файла.

Узор, состоящий из чередующихся черных и прозрачных пикселей, видится как полупрозрачная черная область, наложенная на большую область (рис. 4). Это напоминает технику создания полутоновых изображений струйными принтерами: они наносят решетку из крохотных черных точек, заставляя глаз поверить, будто он видит серую картинку.

Поскольку пиксели больше точек, наносимых струйными принтерами даже с самой низкой разрешающей способностью, то при пристальном взгляде структура узора становится видна, так что иллюзия несовершенна. Но она все же достаточно хороша, поэтому этот узор можно считать частью дизайна (рис. 5). Чтобы увязать его с остальной частью страницы, Дэвидсон еще подчеркнул заголовки в основном тексте пунктирными линиями. Это тонкий ход, призванный придать всей работе стилистическое единство, пусть даже она выглядит не совсем так, как задумано.

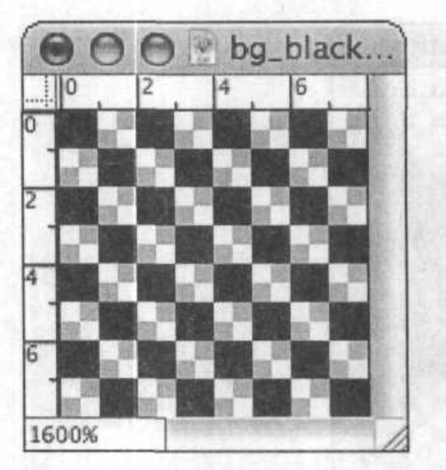


Рис. 4 ✦ Увеличенный GIF, представляющий клетчатый узор, который имитирует 50%-ную прозрачность черной области

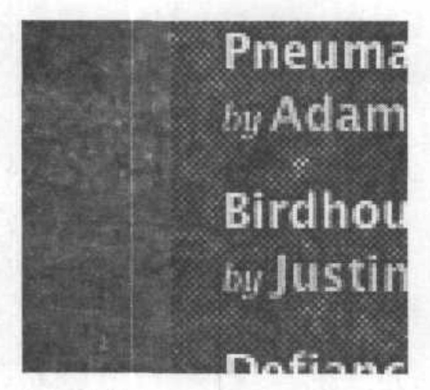


Рис. 5 ✦ Перекрытие «клетчатым» GIF

Специальные приемы

Благодаря примененным Дэвидсоном трюкам с фиксированным позиционированием можно почти с уверенностью утверждать, что пользователь потратит время на прокрутку страницы, просто чтобы посмотреть, как это выглядит. Заметив статуэтку Будды на уступе, он, сам того не сознавая, скользнет взглядом по материалу, расположенному выше и ниже ее.

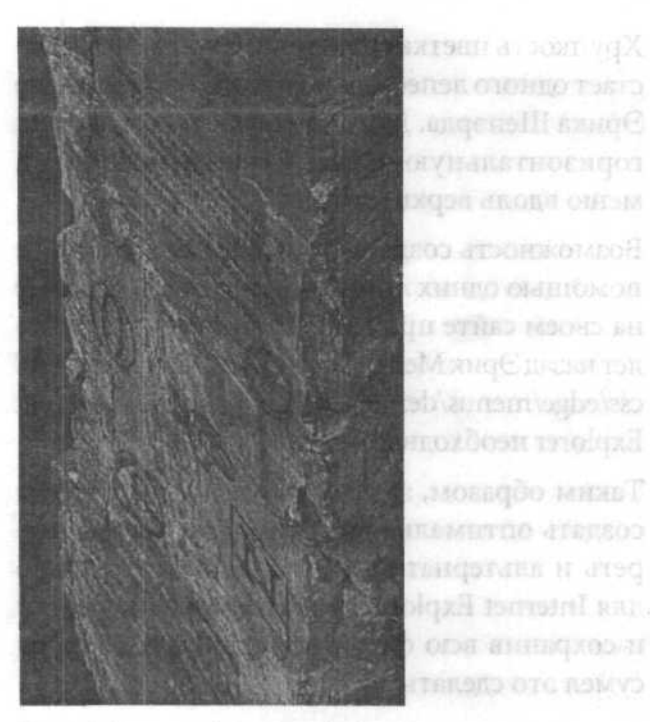


Рис. 6 ✦ Логотипы браузеров в виде рун, высеченных на поверхности утеса

Наградой тому, кто доберется до нижней части страницы, будет приятный сюрприз: логотипы основных браузеров, которые кажутся высеченными на поверхности утеса вместе с фрагментом CSS-кода (рис. 6).

Эта дополнительная графика – необычный прием, поскольку многие дизайнеры предпочитают размещать замысловатые детали в верхней части страницы, так как именно там на них, скорее всего, обратят внимание. Нарушение традиции, выражающееся в том, что автор поместил тематически значимые элементы в нестандартном месте, побуждая зрителя отыскивать их, – это творческий подход, знаменующий неординарность работы.

Судите сами

Хотя не имеет смысла применять хитрые CSS-эффекты, которых большая часть пользователей не увидит, тем не менее для экспериментирования очень важна возможность селективно понижать сложность дизайна, чтобы во всех браузерах он выглядел приемлемо.

Раздвигание границ дозволенности CSS-эффектов требует от дизайнера некоторого азарта, но, к счастью, можно сыграть безопасно, не оставив зрителей разочарованными. Не ограничивая свою свободу, надо все же тщательно учитывать уровень поддержки в разных браузерах, и тогда кто-то сможет увидеть все богатство замысла, а остальные не почувствуют себя ущемленными.

Дизайнер **Эрик Шепэрд** (Eric Shepherd)
www.csszengarden.com/088

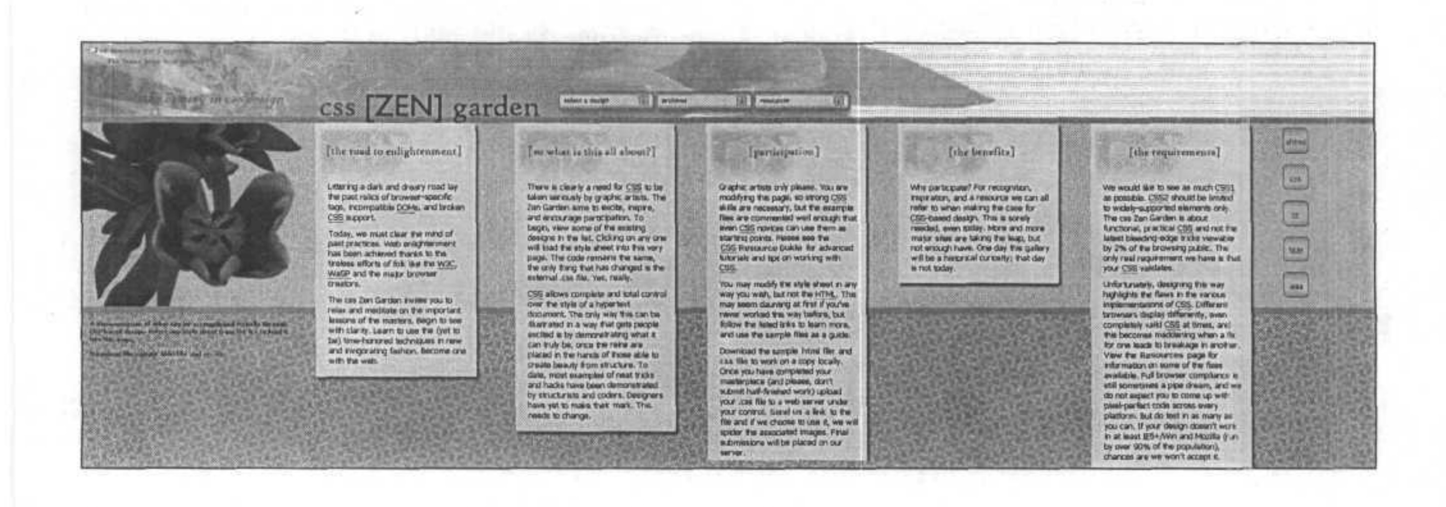
Тюльпан

Реализация ниспадающих меню с помощью CSS и альтернатива для Internet Explorer

Хрупкость цветка тюльпана, у которого недостает одного лепестка, захватила воображение Эрика Шепэрда. Для своей работы он выбрал горизонтальную верстку с ниспадающими меню вдоль верхнего края.

Возможность создавать ниспадающие меню с помощью одних лишь таблиц стилей впервые на своем сайте продемонстрировал несколько лет назад Эрик Мейер (www.meyerweb.com/eric/css/edge/menus/demo), но для работы в Internet Explorer необходимо искать обходные пути.

Таким образом, задача состоит в том, чтобы создать оптимальный дизайн, но предусмотреть и альтернативный вариант специально для Internet Explorer, не утратив целостности и сохранив всю функциональность. Шэперд сумел это сделать.



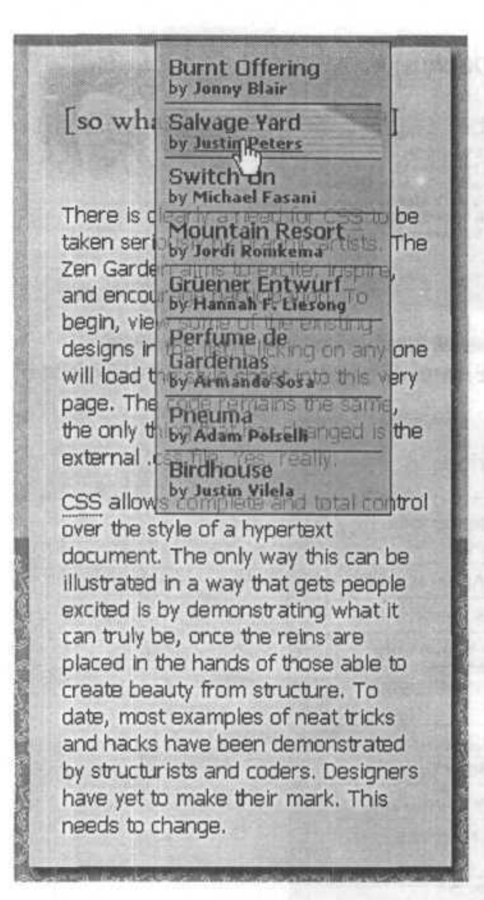


Рис. 1 ✦ Одно из ниспадающих меню в «Тюльпане»

Select a Design:

- Gardenias de Perfume by Armando Sosa
- Pneuma by Adam Polselli
- Birdhouse by Justin Vilela
- Defiance by James Ehly
- Mediterranean by John Whittet
- Austrian's Dark Side by Rene Grassegger
- Invitation by Brad Daily
- Odyssey by Terrence Conley

Рис. 2 ✦ Без стилей разметка отображается любым браузером в виде неупорядоченного списка

Создание меню при помощи одних лишь таблиц стилей

Создавать меню на базе CSS – сплошное удовольствие. Делается это легко, причем с помощью применения различных стилей и способов позиционирования можно добиваться различных результатов. Но самое интересное в том, что не нужно писать ни одной строчки на языке JavaScript (рис. 1).

В приведенном ниже коде показана базовая разметка навигационного меню на основе списка. Это упрощенная версия разметки, примененной в «Тюльпане», разница в том, что вместо нескольких div для разных списков ссылок здесь имеется лишь один.

```
<div id="menu">
<h3>Select a design:</h3>
<ul>
<li><a href="#">Gardenias de Perfume</a> by <a
href="#">Armando Sosa</a></li>
<li><a href="#">Pneuma</a> by <a href="#">Adam
Polselli</a></li>
<li><a href="#">Birdhouse</a> by <a
href="#">Justin Vilela</a></li>
<li><a href="#">Defiance</a> by <a
href="#">James Ehly</a></li>
<li><a href="#">Mediterranian</a> by <a
href="#">John Whittet</a></li>
<li><a href="#">Austrian&#8217;s Dark Side</a>
by <a href="#">Rene Grassegger</a></li>
<li><a href="#">Invitation</a> by <a
href="#">Brad Daily</a></li>
<li><a href="#">Odyssey</a> by <a
href="#">Terrence Conley</a></li>
</ul>
</div>
```

Если посмотреть, как выглядит эта разметка без применения стилей в любом браузере, включая Internet Explorer, то вы увидите обычный неупорядоченный список (рис. 2).

Хотя пользоваться ею, безусловно, можно, но выглядит она не очень красиво. Однако стоит добавить таблицу стилей, как все преображается. Ниже приведена таблица со стилями для тела документа и ссылок.

```
body {
  font-family: tahoma, arial, helvetica, sans-
  serif;
```

```
 font-size: small;
 background-color: #F2F4EE;
}
a:link, a:visited {
 font-weight: bold;
 text-decoration: none;
 color: #998;
}
a:hover {
 color: #887;
 text-decoration: underline;
}
```

На рис. 3 показано, как выглядит эта разметка после применения стилей. Обратите внимание, что даже в Internet Explorer при наведении мыши на ссылку ее текст подчеркивается.

Пока все хорошо, но дальше начинаются сложности. Необходимо изменить значение свойства `display` для элемента `ul` на `none`. В результате список будет скрыт и активируется лишь при наведении на него курсора:

```
ul {
 display: none;
}
```

Добавьте это правило в таблицу стилей, и во всех CSS-совместимых браузерах список исчезнет. Чтобы заставить его снова появиться, нужно добавить динамический псевдокласс `:hover` и задать для него значение свойства `display`, отличное от `none`, в данном случае `block`. Теперь, как только курсор мыши задержится над элементом #menu, элемент `ul` станет видимым. Обратите также внимание на дополнительные стили для элементов `ul` и `li`:

```
#menu:hover ul {
 display: block;
 border: 1px solid #776;
 margin: 5px;
 padding: 5px;
 width: 175px;
 list-style-type: none;
}
li {
 margin: 0;
 padding: 3px;
 border-bottom: 1px solid #887;
}
```

Если протестировать эту таблицу стилей в браузере, поддерживающем псевдокласс `:hover` для произвольных элементов, то при наведении курсора появится меню в виде списка. Именно так оно и выглядит в «Тюльпане» (рис. 4). Увы, в Internet Explorer меню так и остается скрытым, виден лишь его заголовок (рис. 5).

Select a Design:

Gardenias de Perfume by Armando Sosa
Pneuma by Adam Polselli
Birdhouse by Justin Vilela
Defiance by James Ehly
Mediterranean by John Whittet
Austrian's Dark Side by Rene Grassegger
Invitation by Brad Daily
Odyssey by Terrence Conley

Рис. 3 ✦ Применение простых стилей к меню, включая и стиль `:hover`, который поддерживается в Internet Explorer

Select a Design:

Gardenias de Perfume by Armando Sosa
Pneuma by Adam Polselli
Birdhouse by Justin Vilela
Defiance by James Ehly
Mediterranean by John Whittet
Austrian's Dark Side by Rene Grassegger
Invitation by Brad Daily
Odyssey by Terrence Conley

Рис. 4 ✦ В совместимых браузерах, например в Mozilla и Opera, скрытый список появляется при наведении курсора

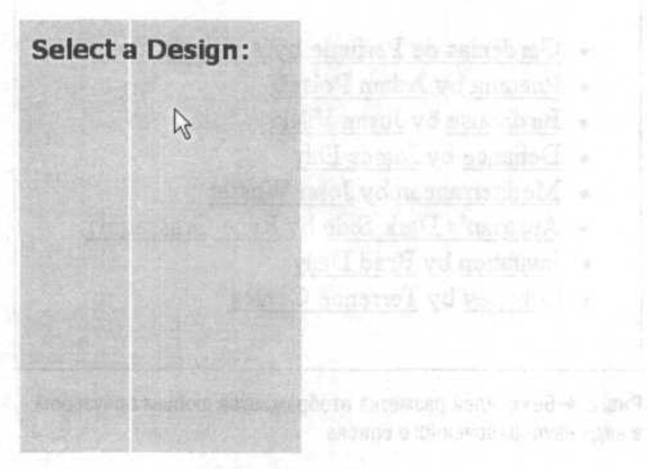


Рис. 5 ✦ При наведении курсора список не отображается, так как Internet Explorer поддерживает псевдокласс `:hover` только для элемента `a`



Ниспадающие меню в «Тюльпане»

Если посмотреть, как выглядит «Тюльпан» во всей красе, то вы увидите все вышеописанное: HTML-списки превращаются в меню, а стили задают поведение при наведении курсора и придают меню красивый внешний вид. Но из-за проблемы в Internet Explorer Шепэрд должен был продумать CSS-код очень тщательно. Задача состояла в том, чтобы дать эффекту проявиться в полной мере в совместимых браузерах, но скрыть эти стили от Internet Explorer, «подсунув» вместо них что-то другое, пусть не такое хитрое, но позволяющее работать с меню, не калеча основную идею дизайна.

Минус на минус дает плюс

Относительно Internet Explorer здесь уместно сделать два замечания. Во-первых, он не позволяет применять псевдокласс `:hover` ни к чему, кроме ссылок. Поэтому эффект наведения курсора воспроизвести не удается, и значит, собственно выпадение меню реализовать таким способом не получится. Во-вторых, как мы знаем, Internet Explorer не понимает некоторых типов селекторов. В частности, речь идет о *селекторе дочерних элементов*, который и составляет главное средство сокрытия стилей от этого браузера. Вот фрагмент из «Тюльпана»:

```
#linkList2 ul > li
```

Internet Explorer не поддерживает также псевдокласс `:first-child`.

```
#linkList2 #lselect li > a:first-child
```

Это правило интерпретируется так: *первая ссылка в любом элементе списка, находящемся внутри элемента с идентификатором* `lselect`, *который является потомком элемента с идентификатором* `linkList2`. К таким элементам будет применен соответствующий стиль, если, конечно, браузер поддерживает подобные селекторы. Коль скоро для Internet Explorer дело обстоит не так, то он просто проигнорирует такое правило.

Отсутствие поддержки для подобных селекторов и псевдоклассов – источник постоянной головной боли при кодировании стилей для Internet Explorer, но иногда эти недостатки можно обратить себе на пользу, чтобы обойти проблему.

Разделение стилей

Чтобы воспользоваться CSS для создания красивых ниспадающих меню и в то же время не отказываться от поддержки Internet Explorer, придется объявить резервные стили, которые Internet Explorer понимает:

```
#linkList2 {
  position: absolute;
  top: 65px;
  left: 750px;
  z-index: 2000;
  height: 150px;
  margin: 0;
  padding: 0;
  width: 1350px;
}
#linkList2 a {
  border: 0;
}
```

Это лишь малая часть полной таблицы стилей, которую можно просмотреть, открыв CSS-файл для дизайна «Тюльпан». Там есть все стили, которые понимает Internet Explorer, а также и дополнительные, распознаваемые лишь браузерами с более полной поддержкой CSS. Эти стили легко отличить по наличию селектора дочерних элементов:

```
#linkList2 #lselect li > a:first-child {
  display: block;
}
html > body #linkList2 li {
  margin: 0;
  padding: 3px;
  border-bottom: 1px solid #887;
}
```

Разделив стили с помощью этого приема, Шепэрд смог реализовать пригодные для работы меню в Internet Explorer (рис. 6). Конечно, они выглядят не так привлекательно, как в других браузерах, но все же не уродливы и для навигации годятся.

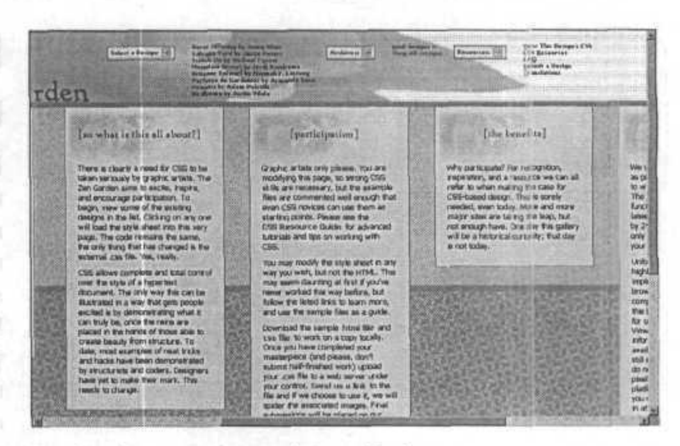


Рис. 6 ✦ Версия меню для Internet Explorer. Не так красиво, как ниспадающие меню, но работать можно

Еще более продвинутый CSS

У реализации меню в «Тюльпане» есть еще один аспект, на который стоит обратить внимание:

```
#linkList2 ul > li:last-child
```

Здесь используется не только селектор дочерних элементов, но и псевдокласс `:last-child`, определенный в спецификации CSS3. Если псевдокласс `:first-child` позволил Эрику выбрать первого потомка в каждом списке меню, то этот псевдокласс дает возможность выбрать последнего потомка и применить к нему свой стиль. Как выясняется, в некоторые браузеры уже встроена рудиментарная поддержка CSS3, так что ему удалось достичь удовлетворительных результатов.

> **СОВЕТ**
>
> Прочитать о работе Дина Эдвардса над Internet Explorer 7 можно на сайте http://dean.edwards.name/IE7/. Прием whatever:hover обсуждается на странице www.xs4all.nl/~peterned/csshover.html.

Но на момент работы над этой книгой спецификация CSS3 еще не была завершена, а вы видели, как плохо некоторые браузеры поддерживают даже имеющиеся рекомендации. Можно ли сделать что-то, кроме поиска обходных путей для Internet Explorer? Вообще-то да, но вам решать, насколько предлагаемые варианты вас устраивают.

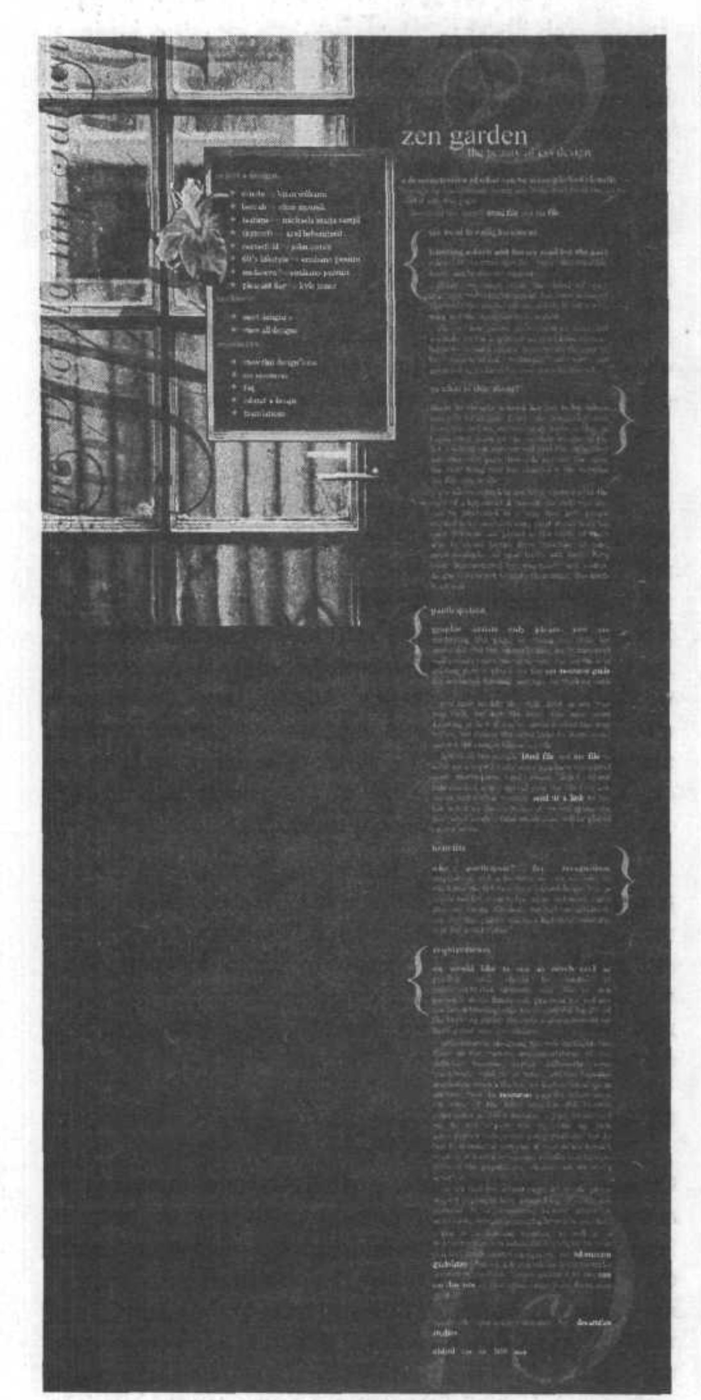
Во-первых, можно по умолчанию реализовывать динамические меню с помощью JavaScript-сценариев. В этом нет ничего порочного, просто CSS позволяет сделать то же самое более элегантно, не задумываясь о том, включил ли пользователь в своем браузере JavaScript или нет.

Другая альтернатива – воспользоваться так называемым *htc-файлом*. Это расширение поддерживает только Internet Explorer. С помощью сценария, записанного в htc-файле, можно изменить поведение самого браузера. Самый известный обходной маневр, реализованный с помощью этой техники, – это проект IE7 Дина Эдвардса (Dean Edwards), в котором с помощью нескольких htc-файлов исправлены некоторые ошибки в Internet Explorer и добавлена поддержка передовых возможностей CSS. Еще один вариант, касающийся псевдокласса `:hover`, – это прием whatever:hover, базирующийся на тех же принципах.

Но надо иметь в виду, что все эти обходные пути официально не поддерживаются, так что пользоваться ими надо с осторожностью, все тщательно взвесив. По своей природе они противоречивы. Ясно, что спецификация должна быть поддержана самим браузером, и некоторые пуристы полагают, что любой трюк или фильтр – неприемлемое решение.

Чему мы научились

От хрупкого тюльпана, давшего начало дизайну, до сложных вопросов организации меню с помощью CSS (которые теоретически решаются элементарно, если бы только браузеры должным образом поддерживали спецификацию) и тонкостей поиска обходных путей и фильтрации. Все это есть в «Тюльпане», являющемся примером изобретательного и вместе с тем ответственного применения таблиц стилей. Да, все дизайнеры с нетерпением ждут того момента, когда необходимость в трюках отпадет, но, пока этот день не настал, нужно учиться, как можно воспользоваться самыми передовыми средствами, предусмотрев в то же время резервный вариант для менее совместимых браузеров. С «Тюльпаном» можно работать и получать эстетическое удовольствие от просмотра в любом браузере.



Дизайнер Патрик X. Лауке (Patrick H. Lauke)
www.csszengarden.com/041

Калитка в мой сад

Простая таблица стилей приводит к впечатляющим визуальным эффектам

Фотограф-энтузиаст Патрик Лауке предпочитает делать фотографии самостоятельно, а не пользоваться готовыми. Не стала исключением и «Калитка в мой сад», присланная на сайт Zen Garden. Лауке использовал фотографию в качестве великолепного фонового изображения.

Кроме фотографирования, Лауке питает пристрастие к концепции *побочного дизайна* (incidental design), которую еще называют *искусством находок* (found art) – использованию случайных предметов в произведениях искусства. В данном случае Лауке отсканировал ржавые железки, валявшиеся рядом с помещением, где он трудится. Они внесли свой вклад в создание ощущения увядания, которым проникнута работа.

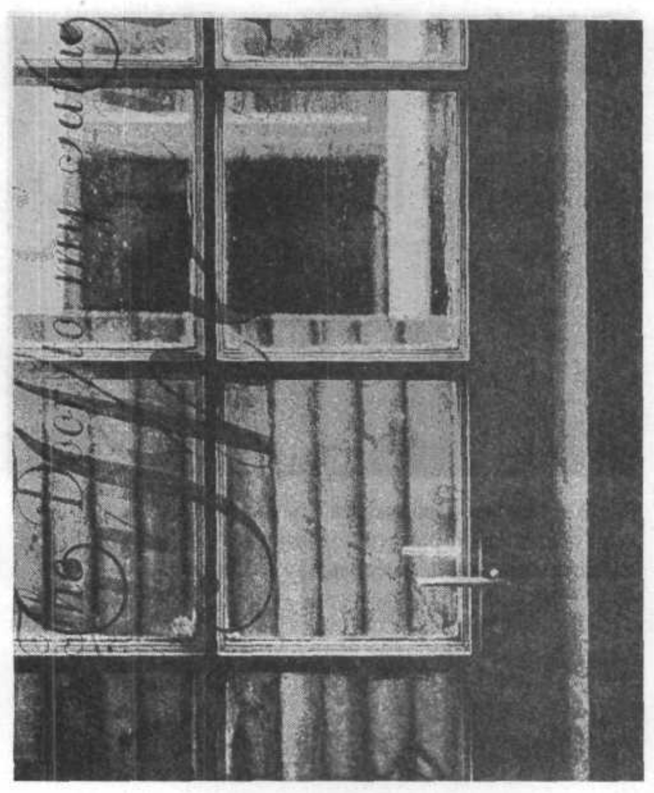


Рис. 1 ✦ Фоновая графика, ассоциированная с элементом body

Важность фона

Хотя и образы, и типография «Калитки в мой сад» играют роль в создании мрачноватого, меланхоличного настроя, но прежде всего этим эффектом дизайн обязан использованию фоновой графики. Конечно, то же самое можно сказать о многих работах на сайте Zen Garden, но именно эта наиболее отчетливо иллюстрирует важность позиционирования фона и его поведения при прокрутке для создания неординарного внешнего вида.

Основной эффект достигается довольно просто и виден в разных браузерах. Лауке привязал фоновое изображение к элементу body и зафиксировал его (рис. 1). Находящаяся справа область контента прокручивается на фоне неподвижной картинки. Работа с изображениями – это краеугольный камень дизайна на основе CSS. Важно понимать, какие в вашем распоряжении есть свойства и как они влияют на внешний вид стилизуемых элементов.

Свойства фона в CSS

У фона есть шесть свойств, включая стенографическое свойство background. Все они перечислены в табл. 1 вместе с необходимыми пояснениями.

Таблица 1 ✦ Свойства фона в CSS

Свойство	Описание
background-color	Определяет цвет фона
background-image	Позволяет разместить картинку в качестве фонового изображения
background-repeat	Определяет, должно ли фоновое изображение повторяться, и если да, то вдоль какой оси
background-position	Позволяет позиционировать изображение относительно прямоугольника, занятого элементом
background-attachment	Определяет, должно ли фоновое изображение прокручиваться вместе с контентом
background	Стенографическое свойство, объединяющее все вышеперечисленные

ПРИМЕЧАНИЕ

Многие дизайнеры определяют не только фоновое изображение, но и цвет фона. Цветами управляет браузер, он показывает их вместе с результатом применения прочих стилей, а для загрузки изображений может потребоваться некоторое время. Поэтому задание одновременно цвета и картинки оставляет у посетителя впечатление более плавной прорисовки страницы.

Добавление фонового изображения и задание цвета фона

Для задания в таблице стилей фонового изображения служит свойство background-image, в значении которого проставляется URL, указывающий на

местоположение картинки. URL задается либо относительно CSS-файла на том же сервере, либо абсолютно – и тогда может указывать на другой сервер:

```
background-image: url(images/my.jpg);
```

Чтобы задать цвет фона, применяется свойство background-color. Значение цвета может задаваться в шестнадцатеричном виде, в сокращенной шестнадцатеричной записи, в виде RGB или одного из поддерживаемых символических названий:

```
background-color: black;
```

Можно также воспользоваться стенографическим свойством background, чтобы сразу задать и цвет, и изображение, и ряд других свойств. Именно так поступил Лауке:

```
body {
  background: #000 url(background.png)
  -30px 0px no-repeat fixed;
}
```

Свойство background-color может также принимать значение transparent, и это активно используется в «Калитке в мой сад». В этом случае сквозь фон элемента проступает то, что находится под ним. В данной работе прозрачность часто используется для отображения ссылок, например:

```
a {
  color: #eee;
  background: transparent;
  text-decoration: none;
}
```

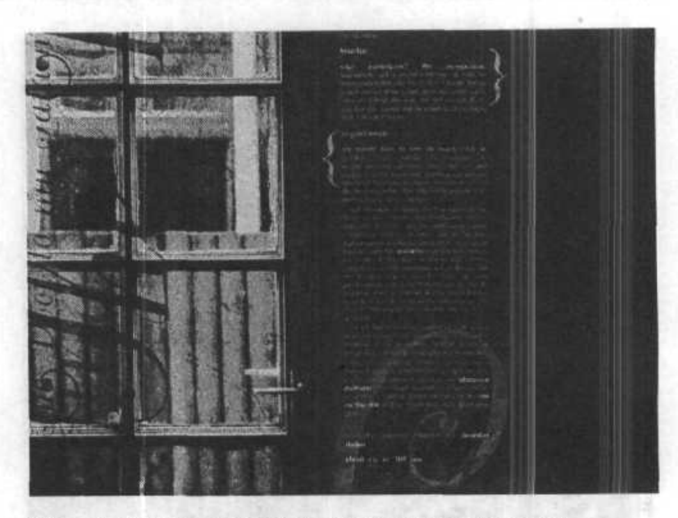


Рис. 2 ✦ Фоновое изображение остается неподвижным, когда контент прокручивается. Именно поэтому можно одновременно видеть конец текста и фотографию

Способы присоединения фоновых изображений

Свойство background-attachment может принимать значения scroll или fixed. По умолчанию предполагается значение scroll, то есть фоновое изображение прокручивается вместе с контентом. Задав значение fixed, можно добиться эффекта, который был нужен Лауке: изображение остается на месте, а прокручивается только контент (рис. 2). Именно таким образом Лауке присоединил фотографию к элементу body:

```
body {
  background: #000 url(background.png) -30px
  0px no-repeat fixed;
}
```

Позиционирование фоновых изображений

Свойство background-position управляет местоположением фонового изображения относительно

> **СОВЕТ**
>
> Задавая значение 0, необязательно указывать единицу измерения. Это замечание относится к любому свойству в CSS. Таким образом, записать приведенное выше объявление можно было бы и так: `background: #000 url(background.png) -30px 0 no-repeat fixed;`

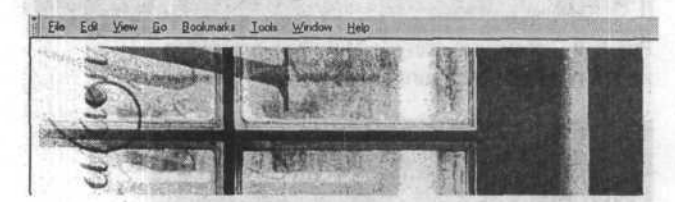


Рис. 3 ✦ Изображение в нормальном потоке

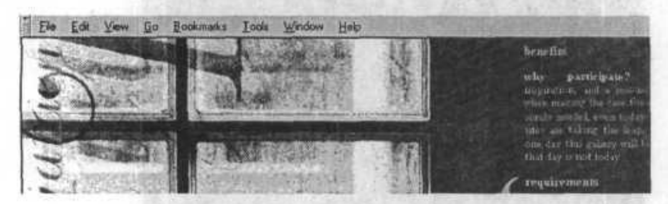


Рис. 4 ✦ Так оно выглядит, если задано отрицательное смещение по горизонтали

родительского элемента. Его значение можно задавать в виде процента, длины или ключевого слова. Значение может состоять из двух позиций. Если заданы обе, то первая определяет положение по горизонтали, а вторая – по вертикали:

```
background-position: 10px 20px;
```

Если задана только одна позиция, то она считается горизонтальной, тогда как вертикальная предполагается равной 50% или `center`. Приведенное выше объявление говорит, что изображение нужно сместить на 10 пикселей по горизонтали и на 20 пикселей по вертикали. Лауке в «Калитке в мой сад» тоже указал положение в пикселях:

```
body {
  background: #000 url(background.png) -30px 0px
  no-repeat fixed;
}
```

Обратите внимание на отрицательное значение `-30px`. Это допускается и означает, что изображение должно быть смещено на 30 пикселей влево, а не вправо, как было бы в случае положительного значения. Задание в качестве позиции значения 0 располагает изображение по вертикали вровень с верхним полем объемлющего элемента, в данном случае `body`. Рисунки 3 и 4 показывают, как выглядит изображение относительно окна браузера без смещения и с отрицательным смещением.

Управление повтором изображения

В «голом» HTML нельзя управлять способом повтора фонового изображения, по умолчанию оно повторяется вдоль обеих осей. CSS же позволяет контролировать это более точно с помощью свойства `background-repeat`.

Это свойство может принимать такие значения: `repeat-x` (повторять только вдоль горизонтальной оси), `repeat-y` (повторять только вдоль вертикальной оси), `no-repeat` (вообще не повторять) и `repeat` (повторять вдоль обеих осей, это режим по умолчанию). Лауке указал в стенографическом свойстве `background` значение `no-repeat`:

```
body {
  background: #000 url(background.png) -30px 0px
  no-repeat fixed;
}
```

Тем самым гарантируется, что браузер не станет заполнять все окно фоновой картинкой. Во всех фоновых изображениях, применяемых в «Калитке в мой сад»,

повторение отключено, хотя иногда встречаются и другие стилистические правила, здесь не показанные.

Варианты верстки

Чтобы получить контейнер шириной 300 пикселей, сдвинутый влево от области просмотра на 470 пикселей, Лауке задал для элемента body отступ 470 px:

```
body {
  padding: 0 0 0 470px;
}
```

Напомним, что в стенографических свойствах padding и margin порядок задания отступов в разных направлениях такой: top, right, bottom, left. Все значения, кроме левого отступа, здесь равны 0 (заметьте, что в этом случае Лауке не указал никакой единицы измерения) (рис. 5).



Рис. 5 ✦ С помощью отступа контейнер отодвинут на 470 пикселей от левого края окна

К этому контейнеру применен стиль, описывающий фоновое изображение, цвет и ширину:

```
#container {
  background: #000 url(bottom_corner.png) no-
  repeat bottom right;
  color: inherit;
  width: 300px;
}
```

С помощью расширения Web Developer для браузеров Mozilla и Firefox вы можете включить показ внешних контуров элементов, а также увидеть информацию об имени класса и идентификаторе для каждого элемента. На рис. 6 показано, как применяются стили для всех элементов внутри раздела #container.



Рис. 6 ✦ Расширение Web Developer позволяет увидеть, как применяются стили к элементам в контейнере #container

Дополнительные стилистические эффекты

В работе «Калитка в мой сад» таблица стилей не содержит почти никаких трюков, поэтому Лауке пришлось решать непростой вопрос об управлении меню и размещении сопровождающего его изображения цветка (рис. 7). Не так-то легко было заставить различные браузеры одинаково показывать окруженное рамкой меню.

Собственно меню не вызывало особых сложностей. Лауке создал контейнер для элемента с идентификатором #linkList и присоединил к каждому элементу



Рис. 7 ✦ Меню и изображение цветка, видны контуры элементов

СОВЕТ

Расширение Web Developer можно бесплатно скачать со страницы www.chrispederick.com/work/firefox/webdeveloper/index.php.

навигации фоновое изображение; в совокупности они и образуют рамку. Но разместить цветок таким образом, чтобы он выступал за рамку и располагался поверх фона, оказалось потруднее.

```
#linkList2>#lselect {
 background: url(flower.png) no-repeat top
left;
 margin-left: -65px;
 padding-left: 80px;
 min-height: 150px;
}
```

Решение, основанное на применении селектора дочерних элементов, работает в браузерах Mozilla, Opera и Safari. Чтобы получить приемлемый результат в Internet Explorer, Лауке воспользовался одним из дополнительных div, имеющихся в разметке, и абсолютно позиционировал изображение цветка:

```
#extraDiv1 {
 position: absolute;
 top: 165px;
 left: 142px;
 background: url("flower.png") no-repeat top
left;
 width: 115px;
 height: 150px;
}
```

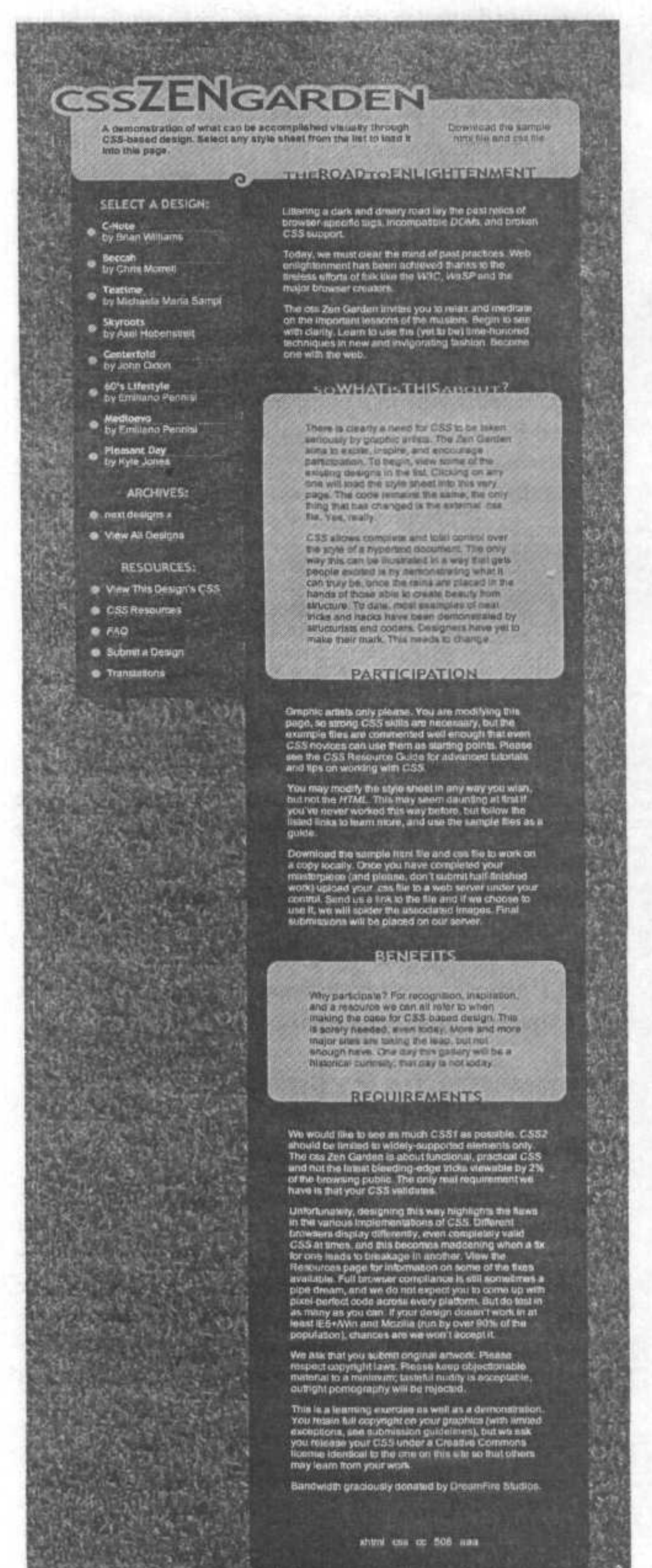
Сейчас совместимые браузеры видят оба стиля. Поэтому Лауке сделал еще один шаг и скрыл последнее правило от браузеров, которые и так уже распознают селектор #linkList2>#lselect:

```
body>#extraDiv1 {
 display: none;
}
```

Voilà! И цветок, и меню видимы и ведут себя так же, как в современных браузерах.

Отличный стиль, простая таблица

Ясно, что своим изяществом «Калитка в мой сад» обязана, прежде всего, уникальным образам, типографии и продуманному размещению элементов. Хотя на первый взгляд дизайн кажется сложным, на самом деле для достижения большинства эффектов используется довольно простая таблица стилей. Чтобы обеспечить совместимость, пришлось применить всего один обходной маневр.



Дизайнер **Патрик Гриффит** (Patrick Griffith)
www.csszengarden.com/063

Упругая лужайка

Для решения визуальных проблем иногда приходится потрудиться

Патрик Гриффит загорелся простой идеей: попробовать новую технику верстки и по ходу дела определить графические элементы. Потребовалось немного времени для осознания того факта, что такой подход «сверху вниз» может привести к хаотичной и неструктурированной процедуре проектирования. Начав с карандаша и блокнота, Патрик быстро перешел непосредственно к работе в терминах CSS. Изображения были добавлены позже, когда основные параметры дизайна определились.

Не существует никакого «единственно правильного» способа верстать страницу на основе CSS. Некоторые дизайнеры сначала воплощают свои идеи в каком-нибудь графическом редакторе типа Macromedia Fireworks, а потом приступают к написанию кода. Другие начинают с таблиц стилей, а позже добавляют графику. Если принять во внимание разнообразие редакторов и графических утилит, то открываются поистине безграничные возможности.



Рис. 1 ✦ Этот сайт (www.alazanto.org), сверстанный Кэвином Дэвисом (Kevin Davis) с фиксированной шириной, прекрасно смотрится при разрешении 1024×768...



Рис. 2 ✦ ... но в окне меньшего размера появляется полоса прокрутки

Бумага или резина?

Проектируя сайт, дизайнер часто должен выбирать между двумя разными принципами верстки: либо задать для области контента фиксированную ширину в пикселях и соблюдать это ограничение для всего сайта, либо позволить странице заполнить все окно браузера, не налагая никаких ограничений по ширине.

Оба подхода имеют названия. Первый часто называют версткой *с фиксированной шириной, с ограничениями* или *застывшей*. Второй – *версткой с переменной шириной, полноэкранной* или *текучей*. Но как бы они ни назывались, это два фундаментальных принципа верстки современных веб-сайтов.

У каждого подхода есть достоинства и недостатки. Один из самых неприятных для графических дизайнеров аспектов Web – это переменная ширина холста. Когда делается дизайн печатных материалов, размеры и пропорции конечного результата заранее известны. Что касается Web, то посетитель сайта может просматривать его как в нераспахнутом окне браузера при разрешении 640×480 пикселей, так и в распахнутом на весь экран окне при разрешении 2048×1536 пикселей. Все промежуточные варианты тоже не исключаются.

Фиксированная ширина

Фиксированная ширина позволяет до некоторой степени контролировать размер холста путем задания точной ширины макета в пикселях. Это идеально в случае, когда нужно сохранить пропорции изображений, которые не могут масштабироваться, и шрифта, который хотя и масштабируется, но непредсказуемо. Но в таком случае пользователь, просматривающий страницу в окне небольшого размера, увидит полосу горизонтальной прокрутки, так как макет может не уместиться в отведенной области (рис. 1 и 2). С другой стороны, пользователи, работающие с большим окном, увидят по бокам пустое пространство.

Переменная ширина

С другой стороны, страница, сверстанная с переменной шириной, заполняет все окно вне зависимости от разрешающей способности экрана (рис. 3 и 4). Теоретически

это идеальное решение, учитывающее различия в размерах окна, но не так-то легко найти оптимальный баланс между шириной 600 и 200 пикселей. Часто приходится в качестве компромисса соглашаться на наименьший общий знаменатель, когда для больших окон разрешается масштабирование, но дополнительно появившееся пространство заполняется неоптимальным образом. Дизайнер Камерон Адамс (Cameron Adams) пробовал решить эту проблему за счет использования JavaScript (www.themainblue.com/experiments/Resolution-Layout/), но CSS сам по себе в этом отношении ограничен.

Текучая верстка проблематична с точки зрения сохранения пропорций, они могут непредсказуемо нарушаться при изменении ширины окна. Соотношение 1,6:1 между размерами фотографии и расположенной рядом с ней закрашенной областью может превратиться и в 2:1, и в 3:1, и даже больше по мере раздвигания окна по горизонтали. Картинки не растягиваются, поэтому остается лишь иметь эту проблему в виду и избегать слишком строгих требований к пропорциональности.

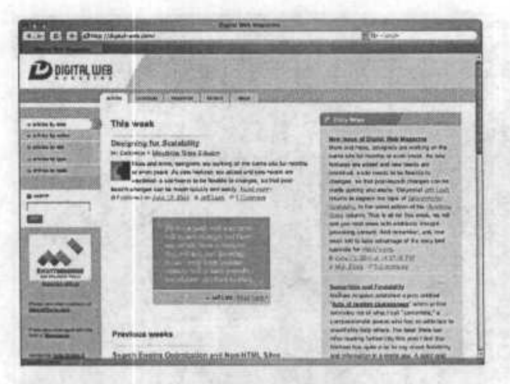


Рис. 3 ✦ Сайт Digital Web Magazine, сверстанный с переменной шириной, прекрасно смотрится при разрешении 1024×768...



Рис. 4 ✦ ... и ничуть не хуже в окне меньшего размера; обратите внимание, что полоса прокрутки не появилась

> **ПРИМЕЧАНИЕ**
>
> Соотношение 1,6:1 означает, что если ширина первого объекта составляет, к примеру, 1,6 см, то ширина второго – 1 см.

Эластичный дизайн

Ну хорошо, мы поговорили о разных методах верстки, но ведь эта глава посвящена специальным эффектам, не так ли? Именно так, в «Упругой лужайке» продемонстрирован эффект «эластичного дизайна», который, не являясь ни версткой с фиксированной, ни версткой с постоянной шириной, представляет собой некоторую смесь из того и другого.

Теория, стоящая за методом Гриффита, выглядит вполне логично. Шрифт, размеры которого заданы в относительных единицах, скажем, `%` или `em`, может становиться больше или меньше в зависимости от размеров окна браузера. Ну а если размеры различных блочных элементов задавать в тех же единицах, то не должен ли масштабироваться *сам дизайн*? Именно в этом и состоит суть эластичного дизайна.

> **ПРИМЕЧАНИЕ**
>
> Гриффит написал статью о своей технике эластичного дизайна и опубликовал ее в январе 2004 года на сайте A List Apart (www.alistapart.com/articles/elastic).



Рис. 5 ✦ С уменьшением размера шрифта главная колонка становится уже



Рис. 6 ✦ При небольшом увеличении размера шрифта увеличивается и ширина основной области



Рис. 7 ✦ При еще большем увеличении размера шрифта основная область продолжает расширяться, а поле над верхним заголовком становится выше

В следующем фрагменте кода показан пример использования относительных единиц в «Упругой лужайке». Ширина элементов, отступов, полей и т.д. выражена в единицах em, а не px, поэтому все масштабируется вместе со шрифтами. На рис. 3–7 видно, как изменяются пропорции при изменении размера окна браузера.

```
body {
  padding: 0 0 6em 0;
}
#container {
  width: 48em;
}
#preamble, #supportingText {
  width: 32em;
  margin-left: 16em;
}
```

Так просто? Всего-то и надо задавать ширину и высоту в единицах em и можно ни о чем больше не беспокоиться? Не совсем. Поскольку размеры картинок фиксированы, масштабировать их вместе с текстом и сохранить пропорции не удастся. Переход к эластичному дизайну требует пересмотра некоторых подходов к проектированию сайтов.

Есть два способа приспособить картинки фиксированной ширины к эластичному контексту. В первом применяется повторение фоновых изображений, во втором – усечение.

Повторение

По умолчанию в CSS фоновое изображение повторяется по горизонтали и вертикали и заполняет всю область элемента. Обычно бывает полезно отключить повтор в обоях или хотя бы в одном направлении, но эластичный дизайн будет выглядеть лучше, если изображение повторяется: тогда, как бы велики ни оказались размеры элемента, фон заполнит его целиком, не оставляя пустот вне области изображения.

В «Упругой лужайке» весь элемент body замощен повторяющимся изображением зеленой травки. Элементы #linkList2, #preamble и #supportingText

заполнены вариантами той же картинки, но других цветов (рис. 8). В сочетании они создают эффект зеленого фона, поверх которого расположены прозрачные колонки других оттенков, содержащие текст и ссылки.

Можно было бы успокоиться, придав каждой колонке стандартную прямоугольную форму, но Гриффит решил расширить концепцию эластичного дизайна, добавив закругленные уголки.

Однако не каждое изображение можно повторить. Если только все четыре края картинки не окрашены в один и тот же цвет, то нужно обращать особое внимание на стыки. На рис. 9 показан небольшой фрагмент необработанного изображения травы, отчетливо видны горизонтальные и вертикальные линии на стыках повторяющихся квадратиков.

Избежать этого можно с помощью небольшой корректировки изображения в Adobe Photoshop. Фильтр Offset (**Filter → Other → Offset**) позволяет имитировать выдвижение картинки за границы холста и появление с другой его стороны, так что сразу становится видно, что произойдет на стыке.

На рис. 10–12 показан процесс совмещения цветов травы, а на рис. 13 окончательный результат. Конкретный способ выбирайте сами – можете взять инструмент Clone Stamp или Healing Brush, чтобы клонировать участки нестыкующихся областей для закрашивания швов, или воспользоваться рассмотренным в главе 4 методом растушевки или какой-либо другой техникой, наиболее подходящей для имеющегося изображения.

Усечение

Приводя в своей статье Elastic Design цитату из онлайнового дневника Хайко Хебига (Haiko Hebig) (www.hebig.org) (рис. 14 и 15), Гриффит отмечает, что другой подход к эластичности связан с использованием изображений, допускающих переменное усечение. По мере увеличения или уменьшения доступной области становится видна большая или меньшая часть изображения.

Есть два простых способа организовать такого рода эластичность: применить свойство `overflow: hidden` к контейнеру изображения, для которого задана ширина в процентах, или же связать с элементом переменной ширины фоновое изображение. Картинку шириной 600 пикселей можно вложить в `div`:

```
<div class="wrapper">
 <img src="skyline.jpg" width="600"
height="200" alt="Skyline" />
</div>
```

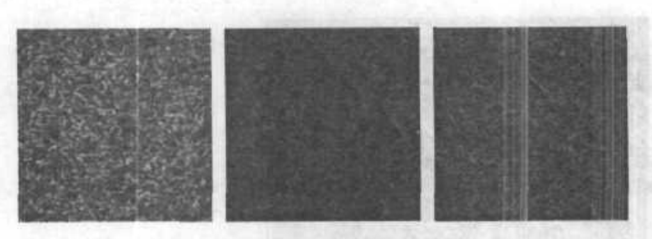


Рис. 8 ✦ Изображения травы, которыми мостится «Упругая лужайка»: ярко-зеленое для основного фона, коричневое для боковой колонки и темно-зеленое для главной области контента



Рис. 9 ✦ При повторении видны стыки

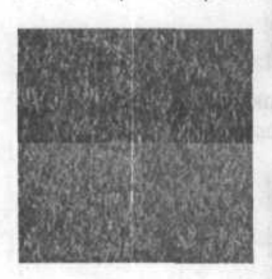


Рис. 10 ✦ На одной смещенной плитке видно несоответствие цветов на стыках

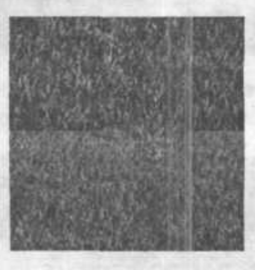


Рис. 11 ✦ В результате клонирования примыкающих областей шов закрашивается

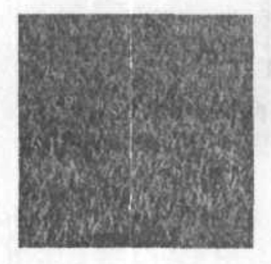


Рис. 12 ✦ Окончательный вид плитки

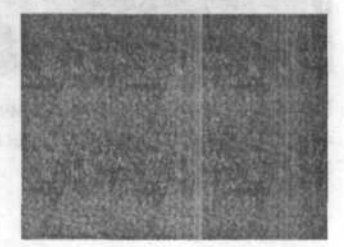


Рис. 13 ✦ Теперь стыки гораздо менее заметны

ПРИМЕЧАНИЕ

Цветной вариант рис. 8–13 см. на вкладке.



Рис. 14 ✦ В онлайновом журнале Хейко Хебига видна большая или меньшая часть расположенной слева фотографии в зависимости от имеющегося пространства. Здесь показано, как выглядит изображение при разрешающей способности 800×600



Рис. 15 ✦ Та же фотография при разрешающей способности 1100×600. Обратите внимание, что ширина фотографии пропорционально увеличилась

СОВЕТ

Свенд Тофте (Sven Tofte) предложил нестандартный, но работающий прием, чтобы компенсировать отсутствие поддержки свойства `max-width` в Internet Explorer 6 (www.svendtofte.com/code/max_width_in_ie).

Далее, если для этого `div` указать ширину, выраженную в процентах, то часть изображения можно скрыть, задав свойство `overflow: hidden`:

```
#wrapper {
 width: 10em;
 overflow: hidden;
}
```

Вместо этого можно связать изображение непосредственно с `div`, указав его в качестве фонового:

```
#wrapper {
 width: 10em;
 background: url(skyline.jpg) no-repeat;
}
```

Если изображение повторяемо, то можно опустить атрибут `no-repeat` и вымостить им элемент в случае, когда размер `#wrapper` начинает превышать исходный размер изображения.

Оттолкнулись – и вперед

Эластичный дизайн – это интересный новый метод верстки веб-страниц, но будьте осторожны: поскольку пропорции дизайна изменяются независимо от размеров окна браузера, то может получиться, что страница окажется больше окна, а это может привести к усечению части изображения, появлению полосы прокрутки и другим неожиданным вещам. В какой-то мере справиться с этим помогает свойство `max-width`:

```
#container {
 max-width: 700px;
}
```

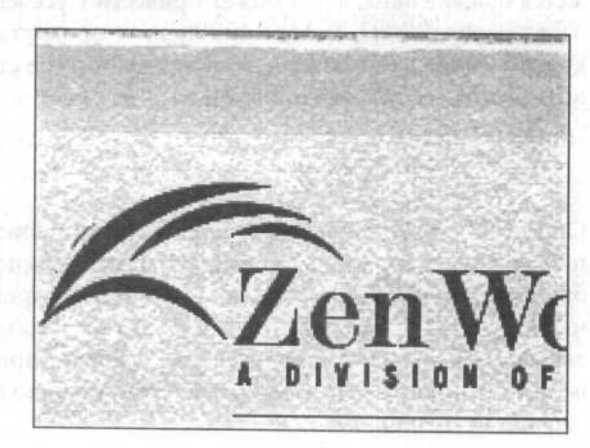
Однако Microsoft Internet Explorer 6 для Windows не поддерживает свойство `max-width`. Имеются способы обойти это ограничение, но все они несовершенны. Причуды браузеров могут побудить к отказу от экспериментов, но что же поделать – освоение нового всегда проходит с переменным успехом, само по себе это не так уж плохо.

Придумывание новых способов работы в Web может окончиться как успехом, так и неудачей, важен конечный итог. «Упругая лужайка» – это блестящий пример того, как эксперименты одного дизайнера могут принести пользу всем.

Живая изгородь, с. 254



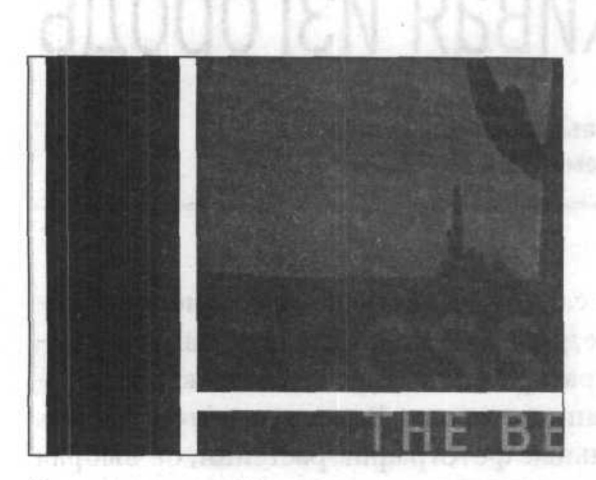
Радио Zen, с. 260



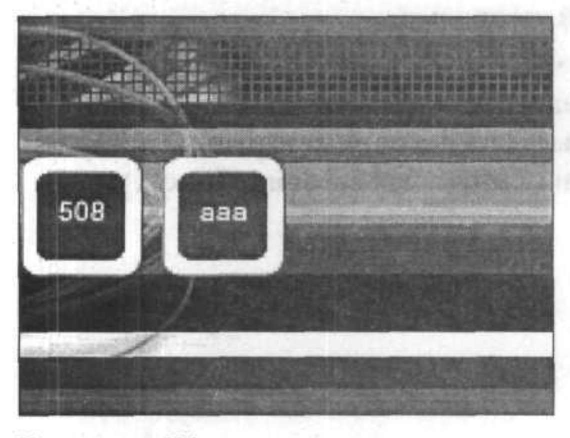
Корпорация ZenWorks, с. 272



Открытое окно, с. 278



К югу от границы, с. 266



Мнемоника, с. 284

7

Реконструкция

Можно сесть и завести разговор о теории, которая всем нам нравится, но если не засучить рукава и не заняться самостоятельно сборкой головоломки, которую представляет собой дизайн на основе CSS, то все эти рассуждения останутся туманными и беспочвенными.

Чтобы вы могли попрактиковаться, мы отобрали шесть вариантов дизайна CSS Zen Garden и проанализировали процесс создания таблицы стилей для каждого из них. Пройдите этот путь с нами, и вы станете лучше понимать процедуры и методы, применяемые в современном дизайне на основе CSS.

Дизайнер **Кевин Мирс** (Kevin Mears)

www.csszengarden.com/031

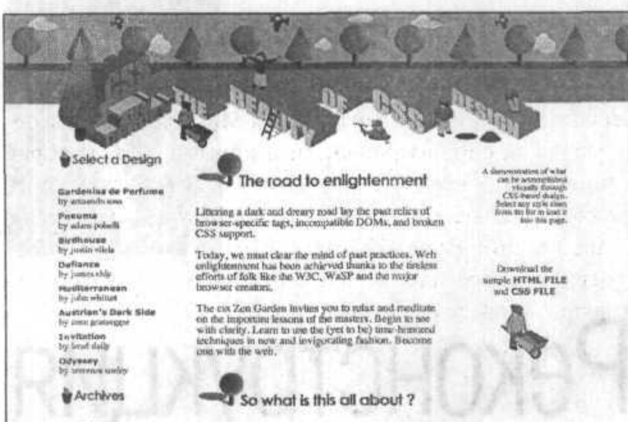
Живая изгородь

Фоновые картинки сливаются с переменным передним планом

Тема сада нашла отражение во многих вариантах дизайна Zen Garden, но дизайнер и иллюстратор Кевин Мирс подошел к ней с неожиданной стороны. Вместо того чтобы делать детальные фотографии растений, он выбрал стиль мультяшек.

Мирс начал с садовника, везущего тачку, а в результате получился живой веселый сад, сверкающий яркими цветами и огражденный живой изгородью.

Ясно, что для «Живой изгороди» понадобилось много изобразительного материала. Отдельные использованные картинки представлены на следующей странице (рис. 1).



Select a Design

Gardenias de Perfume
Pneuma
Birdhouse
Defiance
Mediterranean
Austrian's Dark Side
Invitation
Odyssey

Archives

next designs »
view all designs

Resources

view this design's css
css resources
faq
submit a design
translations

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample **HTML FILE** and **CSS FILE**

The road to enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

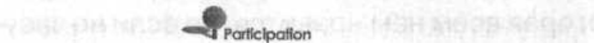
Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So what is this all about ?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.



Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the **CSS Resource Guide** for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample **html file** and **css file** to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. **Send us a link** to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.



Requirements

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the **Resources** page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see **submission guidelines**), but we ask you release your CSS under a Creative Commons license identical to the **one on this site** so that others may learn from your work.

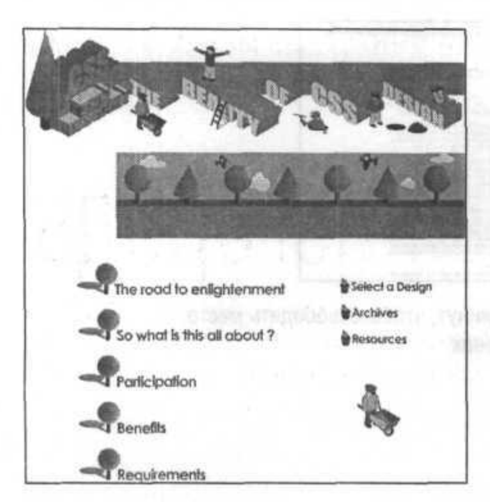


Рис. 1 ✦ Все изображения, использованные в «Живой изгороди»

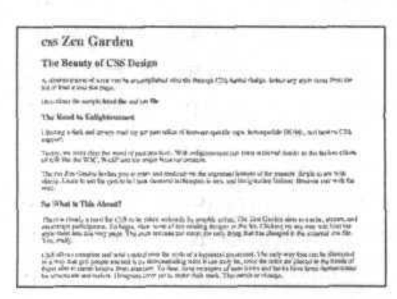


Рис. 2 ✦ Колонка центрирована, но больше пока ничего нет

Очертим границы

Прежде чем включать картинки в дизайн, нужно задать несколько основных параметров. Мирс воспользовался версткой с фиксированной шириной, хотя изображение в заголовке повторяется и заполняет всю ширину окна. Центральная часть страницы ограничена областью шириной 732 пикселя. Раздел `#container` содержит все остальные элементы, поэтому ограничения применяются именно к нему:

```
body {
 margin: 0;
 padding: 0;
 color: #888;
 text-align: center;
}
#container {
 position: relative;
 padding: 0;
 width: 732px;
 margin: 0 auto;
 text-align: left;
}
```

Ширина раздела `#container` установлена равной 732 px, а объявление свойства `margin` центрирует его по горизонтали, поскольку значение `auto` отводит под поля все оставшееся пространство за вычетом ширины самого контейнера. Поскольку для элемента `body` свойство `text-align` установлено в `center`, то для выравнивания содержимого `#container` на левую границу нужно отменить это правило и задать значение `left`. Internet Explorer для Windows версий 5.0 и 5.5 не интерпретирует описанные выше значения свойства `margin` как должно, поэтому и применен этот простой обходной прием, чтобы расположить `#container` по центру страницы. Добавив еще стили для цветов ссылок, мы увидим страницу, изображенную на рис. 2.

Расположение колонок

Следующий шаг – скрыть различные заголовки, чтобы они не портили верстки в колонку.

```
h1 span, h2 span, h3 span, #quickSummary p.p
span {
 display: none;
}
```

Основная область контента уже определена, так что можно переходить к трем расположенным внутри нее колонкам. Разметка сайта CSS Zen Garden затрудняет реализацию того, что задумал Мирс: центральная колонка включает разделы #preamble и #supportingText, но не существует отдельного элемента, который заключал бы оба эти раздела и к которому можно было бы применить стиль. Поэтому к #preamble и #supportingText придется применять стили по отдельности, чтобы визуально они выглядели как часть одной колонки:

```
#preamble {
 position: relative;
 left: 195px;
 padding-bottom: 0px;
 width: 400px;
}
#supportingText {
 position: relative;
 left: 195px;
 padding-bottom: 0px;
 border-bottom: 2px solid #363;
 width: 400px;
}
```

Для каждого элемента задана ширина 400 пикселей, а за счет относительного позиционирования они отодвинуты от левой границы на 195 пикселей, вследствие чего и образуют центральную колонку. Того же эффекта можно было бы добиться, задав значение свойства margin-left, но в оригинальном CSS-файле сделано именно так, как написано выше. На рис. 3. показано, как теперь выглядит страница.

Следующий шаг – разместить в нужном месте боковой контент. Сейчас список ссылок, который должен оказаться слева, находится в самом низу страницы; для перемещения его наверх применяется абсолютное позиционирование:

```
#linkList {
 position:absolute;
 top: 200px;
 left: 0;
 width: 190px;
}
#quickSummary p.p1 {
 position: absolute;
 top: 12em;
 right: 15px;

 width: 120px;
 margin: 75px 0 0 580px;
 font-size: 80%;
 text-align: right;
}
#quickSummary p.p2 {
 position: absolute;
 top: 25em;
```



Рис. 3 ✦ Текст сдвинут, чтобы освободить место для боковых колонок

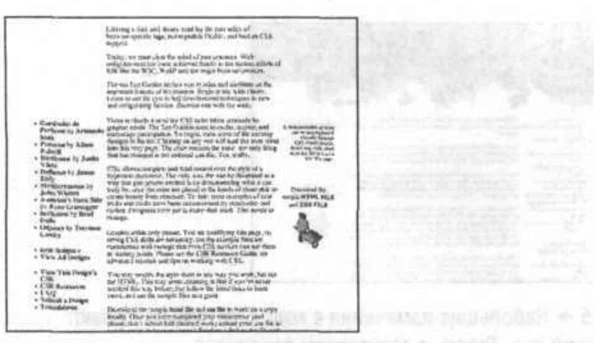
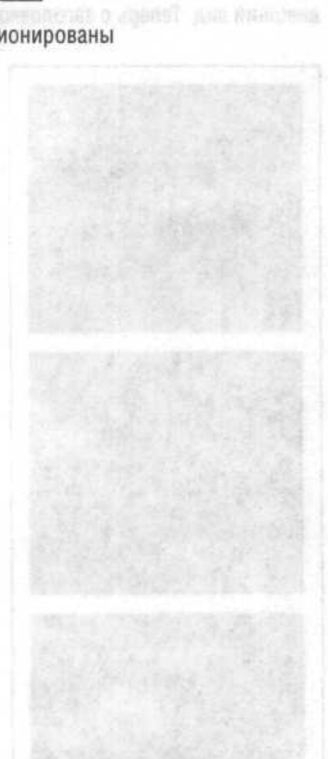


Рис. 4 ✦ Все три колонки позиционированы



```
  right: 15px;
  width: 120px;
  margin: 0;
  font-size: 93%;
  text-align: center;
  padding-bottom: 90px;
  background: url("barrow.gif") no-repeat
bottom;
}
```

Тут сделано много всего. Элемент #linkList получил ширину, перемещен наверх и придвинут к левой стороне контейнера #container. Оба абзаца внутри раздела #quickSummary точно также придвинуты к правой стороне контейнера (можно было бы вместо этого позиционировать общего родителя обоих – элемент #quickSummary - и тем самым избавиться от одного лишнего правила). В данный момент страница выглядит так, как показано на рис. 4.

Хитрости с картинками

Теперь, когда страница начала обретать желаемый вид, рассмотрим, как заполняются графические области. Начнем с заголовка. Он состоит из повторяющейся группы деревьев на заднем плане и расположенного поверх них графического текста. Проще всего добавить повторяющееся фоновое изображение:

```
body {
 margin: 0;
 padding: 0;
 background: #FFF url("bg_tree.gif") repeat-x
left top;
 color: #888;
 text-align: center;
}
```

К имевшемуся ранее правилу body добавлены фоновая картинка и цвет фона. Включив в объявление свойства background значение repeat-x, мы тем самым утверждаем, что картинка должна повторяться только по горизонтали, а это нам и нужно.

К основному заголовку h1 страницы присоединяется еще одна, на этот раз неповторяющаяся картинка, расположенная поверх уже присоединенного к элементу body фонового изображения. Напомним, что для каждого span внутри всех заголовков на странице задано свойство display: none, поэтому текст скрыт и уступает место графике, которую мы сейчас и добавим. Но сначала надо задать для h1 правильные размеры:

```
#pageHeader h1 {
 height: 200px;
 background: url("nutitle.gif") no-repeat;
 padding: 0;
}
```

Поскольку этой картинке нужна длинная горизонтальная область, то элементу h1 разрешено расширяться и заполнять весь контейнер (то есть элемент #container, для которого ранее была задана ширина 732 px); подкорректировать нужно только высоту height. Обратите внимание, что верхний край центральной колонки тоже сместился вниз; поскольку h1 появляется в разметке раньше, то заданная для него высота влияет на положение последующих элементов. Боковые колонки при этом не затрагиваются, так как они позиционированы абсолютно и, значит, изъяты из нормального потока документа. Два этих мелких изменения радикально меняют внешний вид страницы (рис. 5).

Поскольку фон, ассоциированный с элементом body, фиксирован, а положение заголовка переменно, то нельзя заранее сказать, перекроет ли заголовок (например, фигурка с раскинутыми руками) деревья или небо. Наблюдать этот эффект можно, меняя размер окна браузера, – вы увидите, что положение фигурки относительно фона изменяется. Но при сохранении изображения, связанного с заголовком, верхняя половина была сделана прозрачной, поэтому оно прекрасно выглядит на фоне заднего плана. Каков бы ни был размер окна браузера, оба изображения сочетаются без видимых «швов» (рис. 6).



Рис. 5 ✦ Небольшие изменения в коде CSS сильно меняют внешний вид. Теперь с заголовком покончено

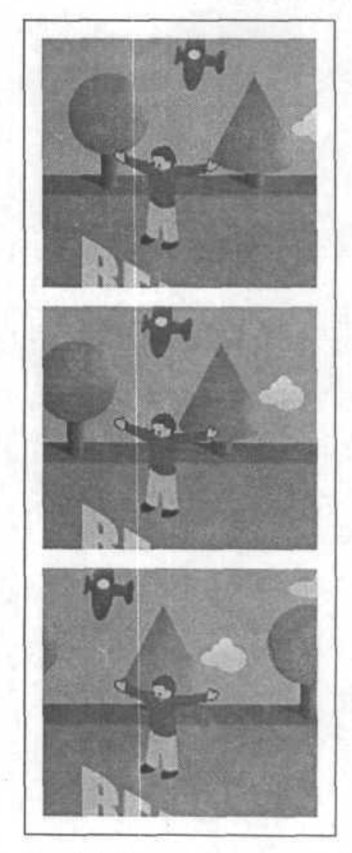


Рис. 6 ✦ Фигурка на переднем плане сливается с фоновым изображением даже тогда, когда ее положение изменяется вследствие изменения размера окна браузера

Форматирование

Поскольку в окончательном дизайне по-прежнему используется шрифт, подразумеваемый браузером по умолчанию, то нужно внести еще несколько мелких дополнений, начав с центральной колонки:

```
#supportingtext p {
  line-height: 1.5em;
}
#preamble p {
  width: 350px;
}
#footer {
  background: D9D98B;
  color: #fff;
  padding: 10px 20px;
```

СОВЕТ

В отличие от многих других свойств CSS, в значении свойства `line-height` необязательно указывать единицу измерения: `line-height: 1.5;` – это корректная запись, означающая то же самое, что `line-height: 1.5em`.



Рис. 7 ✦ Почти все. Осталось только разобраться с заголовками

```
 border-top: 1px solid #363;
 font: 85% Verdaba,Arial,Sans-serif;
 text-align: center;
}
#footer a:link, #footer a:visited {
 padding: 0 5px;
 font-weight: normal;
}
```

Первое правило уменьшает расстояние между строками (свойство `line-height`). Затем для блока `#preamble` устанавливается ширина 350 px, вероятно, для того чтобы слегка увеличить промежуток между этим блоком и правой колонкой.

Хвостовик требует чуть больше форматирования. Ранее для элемента `#supportingText` уже была задана зеленая нижняя граница, отображаемая прямо под хвостовиком; теперь мы добавим к самому элементу `#footer` цвет фона `background` и верхнюю границу `border-top`. После этого страница будет выглядеть, как показано на рис. 7.

Завершающие штрихи

Но не забудьте, что необходимо что-то решить по поводу скрытых заголовков:

```
#supportingText h3 {
 height: 70px;
}
#explanation h3 {
 background-image: url("what.gif");
 background-repeat: no-repeat;
}
#linkList h3 {
 height: 41px;
}
#lselect h3 {
 background-image: url("select.gif");
}
```

Мы не привели большую часть стилей для отдельных заголовков, ведь никаких сюрпризов там нет. Сначала задается согласованная высота `height` для всех элементов h3 внутри первых двух разделов, а затем с каждым заголовком ассоциируется фоновая картинка `background-image`, которая придает дизайну завершенный вид.

Дизайнер **Марк ван ден Хойфель** (Marc van den Heuvel)

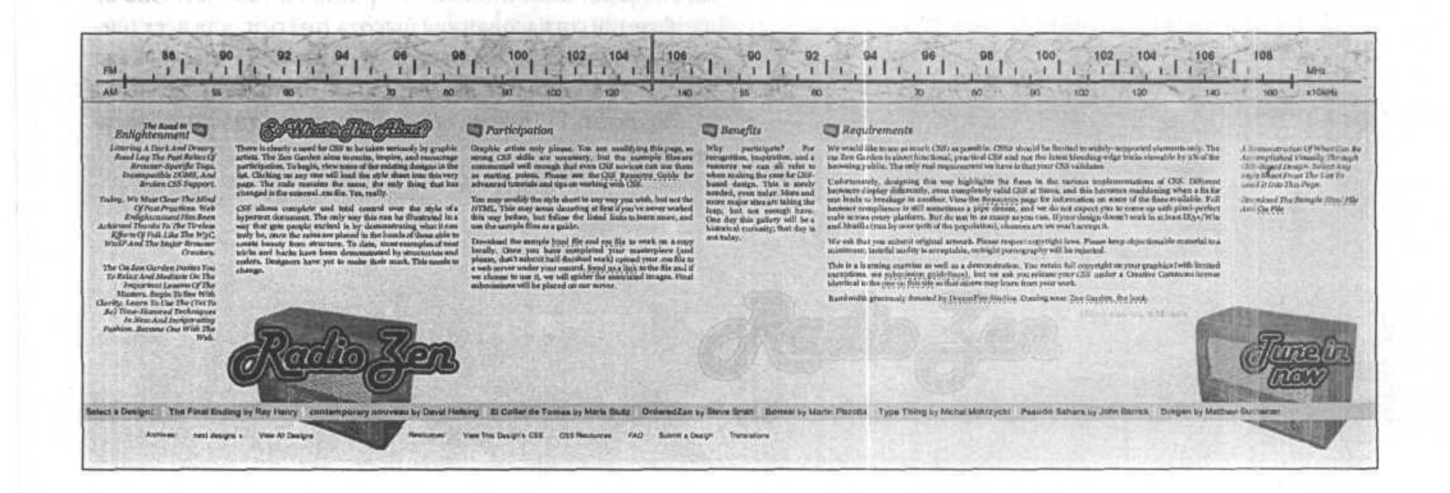
www.csszengarden.com/058

Радио Zen

Шкала настройки в виде фиксированного фонового изображения

Вдохновившись видом своего старенького радиоприемника, Марк ван ден Хойфель решил воспользоваться фиксированным фоновым изображением, чтобы создать образ шкалы настройки частот вместе со скользящим индикатором. Разметка была обусловлена заголовком, а поскольку индикатор перемещается по шкале горизонтально, то появление горизонтально прокручиваемой страницы было почти неизбежно.

Классический облик подчеркнут шрифтом в стиле «ретро», фотографиями радиоприемника, от которого все и пошло, и желтоватой цветовой гаммой, символизирующей составившийся материал.



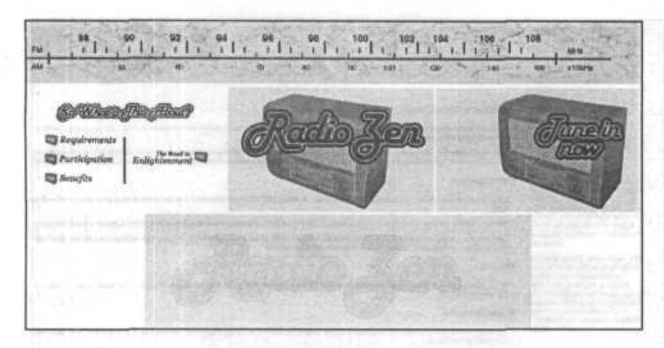


Рис. 1 ✦ Картинки, использованные в «Радио Zen»

Рис. 2 ✦ «Радио Zen» после форматирования текста

> **ПРИМЕЧАНИЕ**
>
> Браузер Microsoft Internet Explorer для Windows поддерживает фиксированное присоединение изображения только к элементу `body`.

Готовим сцену

Если не считать изображений в заголовках, то в работе «Радио Zen» эффект достигается при минимальном использовании графики (рис. 1). Уникальный внешний вид горизонтального макета обеспечивают имеющиеся в CSS средства позиционирования.

Для начала ван ден Хойфель также скрыл текст обоих основных заголовков. Позже они послужат для размещения картинок, а пока пусть не мешаются:

```
#pageHeader h1 span, #pageHeader h2 span {
 display: none;
}
```

Применены и некоторые другие базовые средства форматирования, здесь мы их не показываем, но при желании можно ознакомиться с исходным CSS-файлом (рис. 2). Прежде чем добавлять прокручиваемую шкалу, необходимо расчистить для нее место.

Основные особенности верстки

Для размещения большей части элементов в «Радио Zen» применяется абсолютное позиционирование, следовательно позиционированные элементы изымаются из нормального потока документа. Поскольку оставшиеся элементы ничего не знают о положении начала или конца абсолютно позиционированных элементов, то перекрытие неизбежно. На какое-то время отвлечемся от этой проблемы, сразу расположив несколько элементов так, чтобы их края соприкасались; тогда перекрытия не будет. Соответствующие правила приведены ниже:

```
#quickSummary {
 position: absolute;
 right: 0px;
 text-align: left;
 width: 165px;
 height: 340px;
 top: 95px;
 padding-top: 40px;
}
#preamble {
 position: absolute;
 left: 10px;
 text-align: right;
 width: 165px;
 top: 95px;
}
```

```
#supportingText {
 position: absolute;
 width: 1230px;
 padding: 95px 0 0 185px;
 text-align: justify;
 z-index: 3;
}
#linkList {
 font-family: Arial, Helvetica, sans-serif;
 position: absolute;
 top: 0;
 padding-top: 480px;
 text-align: justify;
 z-index: 1;
}
```

Этот план срабатывает, если размер шрифта в браузере по-прежнему составляет 10 px, как было задано ранее (рис. 3). Но поскольку многие браузеры позволяют пользователю масштабировать текст, то перекрытие все же возможно (рис. 4). Мы еще вернемся к этой проблеме ниже.

Рис. 3 ✦ После применения базового форматирования

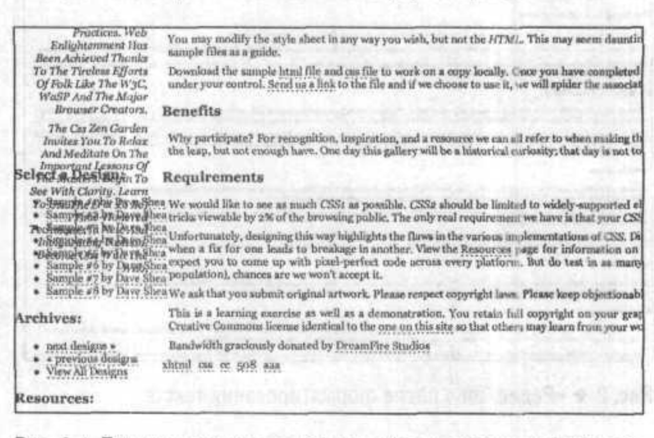


Рис. 4 ✦ При увеличении размера шрифта возможно перекрытие

Организация слоев и присоединение графики

Если вы еще не посмотрели, как оригинальный дизайн прокручивается по горизонтали, рекомендуем сделать это сейчас (рис. 5). Как ни странно, эффект достигается без использования сценариев. В «Радио Zen» изобретательно применены слои и значение `fixed` атрибута `background-attachment`. Поскольку изображение шкалы радиоприемника присоединено к элементу `body` статически, то при любой прокрутке страницы оно остается на месте, а оставшаяся часть страницы прокручивается поверх картинки. В правиле для элемента `body` есть строка, в которой заданы значения свойств `background-image` и `background-position`:

```
body {
 font: 10px Georgia, Arial, Helvetica,
sans-serif;
 color: #353638;
 background: #F8F8EE(schaal.jpg) no-repeat
fixed top center;
 margin: 0px;
}
```

А индикатор шкалы присоединяется к элементу `#supportingText`:

```
#supportingText {
 position: absolute;
 width: 1230px;
 padding: 95px 0 0 185px;
 text-align: justify;
 background: #F8F8EE(naald.gif) top center
no-repeat;
 z-index: 3;
}
```

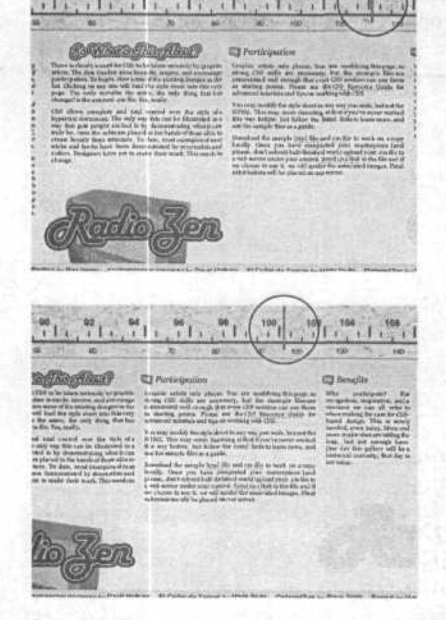


Рис. 5 ✦ При прокрутке страницы индикатор перемещается вдоль шкалы

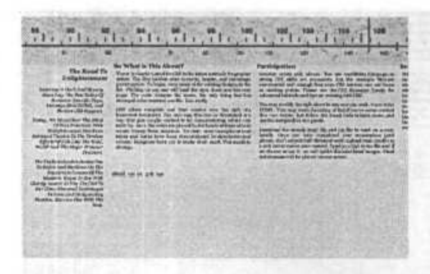


Рис. 6 ✦ Абзацы оказались в горизонтально расположенных колонках

Рис. 7 ✦ При масштабировании текст перекрывается

Теперь должно быть понятно, почему для элемента #supportingText задана такая большая ширина: не просто для того, чтобы в нем поместились горизонтально выстроенные абзацы, но и чтобы гарантированно появилась полоса прокрутки, позволяющая наблюдать эффект перемещения индикатора.

Позиционирование колонок

Пока что текст в теле страницы представляет собой последовательность очень широких абзацев. Исправим это, расположив их горизонтально поперек окна браузера:

```
#preamble {
 position: absolute;
 left: 10px;
 text-align: right;
 width: 165px;
 top: 95px;
}
#supportingText h3 {
 margin: 0 8px 0 3px;
}
#explanation, #participation {
 float: left;
 width: 290px;
}
#benefits {
 float: left;
 width: 150px;
}
#requirements {
 float: left;
 width: 500px;
}
```

Так гораздо лучше (рис. 6). Теперь вернемся к проблеме масштабируемого текста.

Проблема масштабируемого текста

Иногда текст, слишком длинный для отведенной под него области, начинает перекрывать другие элементы, поскольку при абсолютном позиционировании элемент исключается из нормального потока документа. Как это выглядит, можно увидеть, просмотрев страницу с текущими стилями в браузере, поддерживающем масштабирование текста, для которого задана единица измерения px, – например, в Firefox. Попробуйте увеличить размер шрифта и полюбуйтесь, что произойдет со списком вариантов дизайна в нижней части страницы (рис. 7).

Конечно, стили для этой области еще не совсем готовы, но проблема останется, если не позаботиться о ней прямо сейчас. Возможно, вы уже обратили внимание, что элемент #linkList абсолютно позиционирован вверху страницы (свойство top равно 0), но смещен вниз за счет того, что свойство padding-top равно 480 px. Задание такого отступа располагает список вариантов дизайна ниже остальной части страницы на промежуточной фазе верстки страницы, но в конечном счете это неважно, так как оба вложенных в него div все равно будут позиционированы абсолютно, поэтому отступ не сыграет роли:

```
#lselect, #larchives, #lresources {
 position: absolute;
 text-align: left;
 padding: 5px;
}
#lselect {
 width: 1590px;
 top: 46em;
 background-color: #E8E9CF;
}
#larchives {
 width: 300px;
 top: 49em;
 left: 75px;
}
#lresources {
 width: 800px;
 left: 400px;
 top: 49em;
}
```

Поскольку свойство top каждого div задано в единицах em, а не px, то положение верхнего края будет меняться при изменении размера шрифта, что позволит избежать перекрытия. А после применения дополнительных стилей, показанных ниже, список вариантов дизайна и список архивов в нижней части страницы станут, наконец, выглядеть так, как показано на рис. 8.

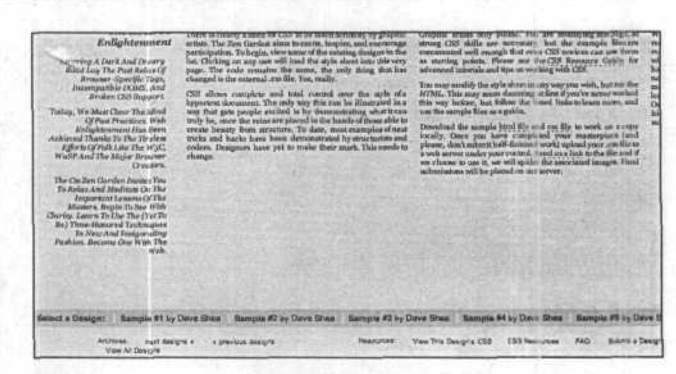


Рис. 8 ✦ Ссылки в хвостовике страницы после применения стилей

СОВЕТ

Применение значений, измеренных в em, в сочетании с абсолютным позиционированием заставляет браузер вычислять положение с учетом размера шрифта. Если же применяются единицы px, то положение задается жестко и не пересчитывается при изменении размера окна браузера, даже если это приводит к перекрытию.

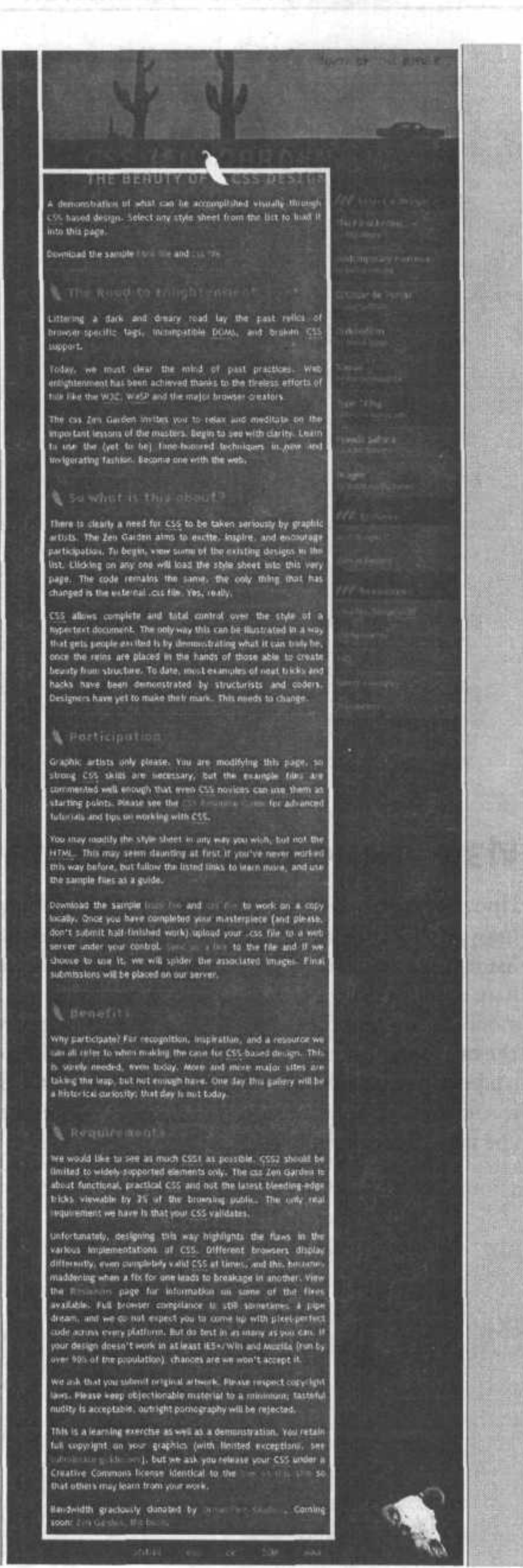
ПРИМЕЧАНИЕ

Мы еще не рассмотрели размещение картинок – в частности, заголовка и двух нижних изображений радиоприемника. Оставляем это в качестве упражнения для читателя. Впрочем, со всеми стилями можно ознакомиться в оригинальном CSS-файле для дизайна «Радио Zen».

```
#lselect ul, #larchives ul, #lresources ul {
 margin: 0px;
 padding: 0px;
}
#lselect h3 {
 margin: 0 10px 0 0;
 color: #555F44;
 font-size: 8pt;
 float: left;
 font-weight: bold;
}
#lselect li {
 #color: #000000;
 list-style-type: none;
 float: left;
 border-left: 4px solid #F8F8EE;
 padding: 0 5px;
}
#lselect a:link, #lselect a:visited {
 border: none;
 text-decoration: none;
 font-size: 8pt;
 font-weight: bold;
}
#lselect a:hover, #lselect a:active {
 text-decoration: underline;
}
#lselect a.c, #lselect a:link.c, #lselect
a:visited.c {
 color: #422222;
 font-weight: normal;
}
```

Изучение возможностей

Простые на первый взгляд свойства CSS при ближайшем рассмотрении могут порождать интересные взаимодействия. Так, фоновые картинки, присоединенные к элементу body с атрибутом fixed, дают любопытный эффект перемещения подвижного контента по неподвижному фону. В «Радио Zen» этот эффект нашел прекрасное применение, но в CSS есть и много других возможностей, открывающих широкое поле для экспериментов.



Дизайнер **Роб Шилдс** (Rob Shields)

www.csszengarden.com/093

К югу от границы

Контент, загнанный в стойло с помощью полей и позиционирования

Большая доза кофеина – это как раз то, что надо, чтобы подстегнуть воображение. В нашем случае идеи посетили Роба Шилдса после приема толики горячего чая. Уж не знаем, что там было намешано, но в результате появилось никак не меньше 50 идей – в виде ключевых слов, описывающих общую тему сайта.

В конце концов Шилдс остановился на третьей идее – теме Юго-Запада. Он не стал заранее ограничивать свое воображение, решив, что найдется место для любой полезной картинки. После чего запустил Adobe Photoshop и работал с графикой, пока не нашел понравившуюся композицию (рис. 1). Немного подправив картинки в Photoshop, Шилдс приступил к кодированию CSS, и в результате получился дизайн «К югу от границы» с четко выраженной темой.

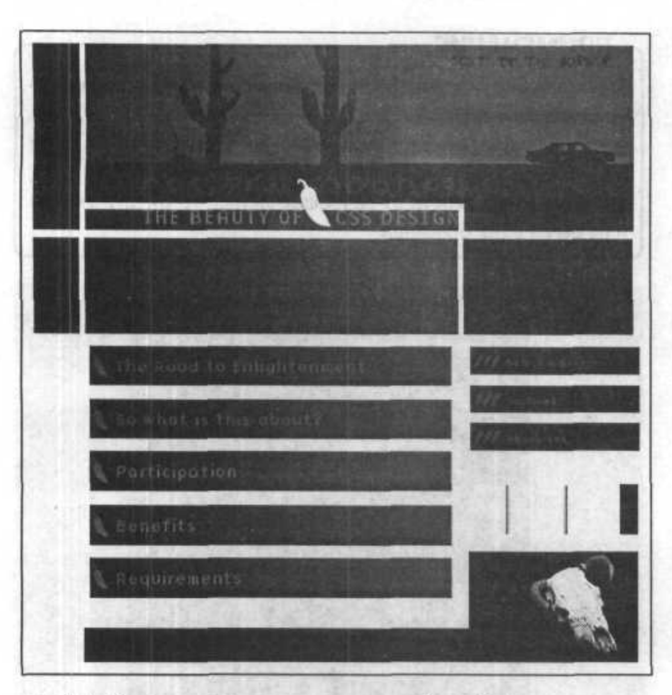


Рис. 1 ✦ Графика, использованная в дизайне «К югу от границы»

Начнем с простого

Один из интересных аспектов работы Шилдса – это выбор цветовой гаммы, на которую его вдохновил вид коричневого костюма в синюю полоску. Он подготовил площадку для работы, задав светло-синий фон в элементе body, светло-коричневый и более темный оттенок синего – для ссылок и белый цвет для основного текста, расположенного на темно-синем фоне:

```
body {
  font: 83%/150%;
  color: #fff;
  background: #EDEDED url(background.gif)
 repeat-y fixed bottom center;
  margin: 0px;
  padding: 0px;
 }
```

Шилдс намеренно задал размер шрифта в процентах, чтобы он мог масштабироваться. Эффект от такого управления шрифтами и фоном показан на следующей странице.

Стилизация контейнера

Задав основные свойства (опущенные здесь фрагменты таблицы стилей можно посмотреть в исходном CSS-файле), можно переходить к стилизации раздела #container. Ниже приведены стили для присоединенной к контейнеру фоновой картинки, полей и ширины:

```
 #container {
  position: relative;
  background: url(bg-central.gif) repeat top
 left;
  margin: 0px auto 0px auto;
  width: 655px;
 }
```

Разумеется, #container позиционирован относительно, а поскольку свойства top и left остались незаданными, то этот элемент остался в нормальном потоке. Тех, кто только приступает к изучению CSS, часто вводит в заблуждение слово «относительное». Кажется, что речь идет о позиционировании относительно какого-то другого элемента. Это естественная, но совершенно неправильная интерпретация.

Относительное позиционирование позволяет разместить элемент, не изымая его из нормального потока. Это совсем не то, что делает браузер с элементами, позиционированными по-другому, например абсолютно. Очень скоро вы отчетливо увидите разницу.

А пока наша цель – создать контейнер, визуально окружающий находящиеся внутри него элементы. После применения стилей к разделу #container (см. оригинальный CSS-файл), можно заняться дополнительной стилизацией содержащихся в нем элементов (рис. 2).

ПРИМЕЧАНИЕ

Термином «нормальный поток» описывается способ обработки браузером непозиционированных элементов. Если удалить все стили из любого варианта дизайна CSS Zen Garden, то текст и заголовки будут располагаться слева направо, сверху вниз, плавая в зависимости от разрешающей способности экрана и размера окна браузера.

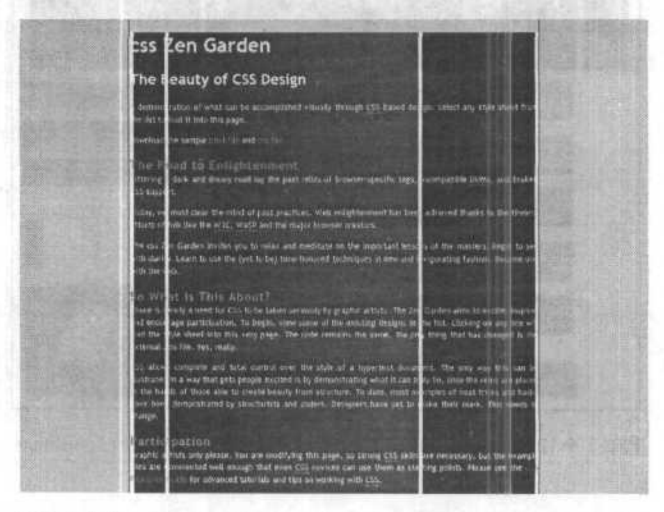


Рис. 2 ✦ Для раздела #container заданы позиция, ширина, поля и фон

Взнуздаем текст

Теперь основной текст следует отодвинуть от краев объемлющего контейнера. Это достигается заданием полей для каждого div внутри элемента #container:

```
#intro {
 margin:0px;
}
#quickSummary, #preamble {
 margin: 0px 200px 0px 68px;
}
#pageHeader {
 margin: 0px;
}
#supportingText {
 margin: 0px 200px 0px 68px;
}
```

При такой установке полей текст занимает нужное положение (рис. 3).

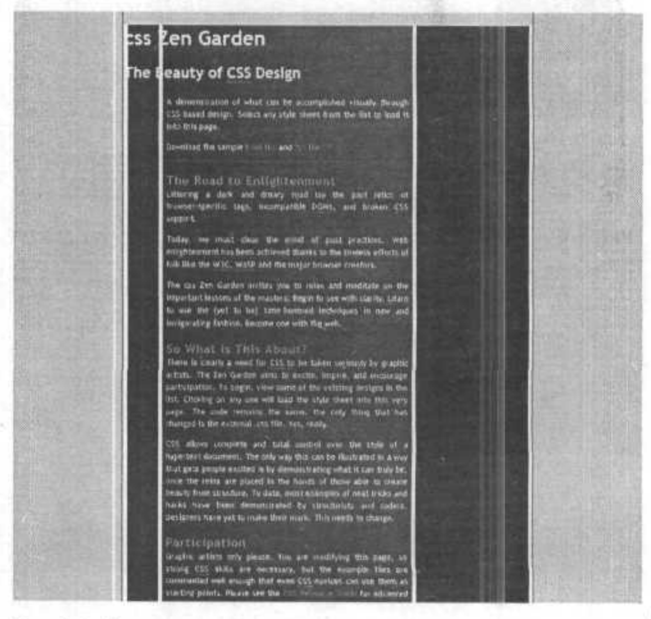


Рис. 3 ✦ После задания полей для основного текста

В шапку их!

Добавление графических заголовков – в особенности для элемента #pageHeader h1 – уравновешивающих текст, приближает дизайн к окончательному виду:

```
#pageHeader h1 {
 margin: 0px 0px 0px 0px;
 background: transparent url(top.gif) no-
repeat top left;
 width: 655px;
 height: 206px;
}
```

На рис. 4 показано, как выглядит страница после присоединения картинки к элементу #pageHeader h1.



Рис. 4 ✦ Присоединение картинки к `#pageHeader` подчеркивает тему и задает общий тон дизайну

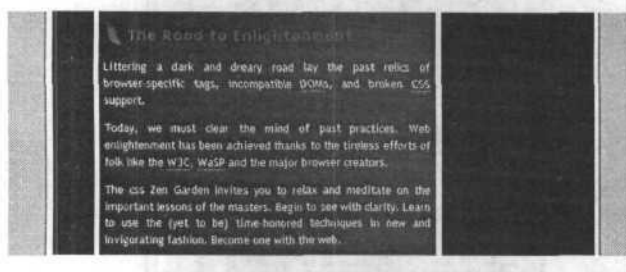


Рис. 5 ✦ На картинках в заголовках изображен стручок перца чили, подчеркивающий тему Юго-Запада

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

Рис. 6 ✦ Рамка, добавленная к разделу `#benefits`, разделяет соседние фрагменты текста

Еще несколько достойных внимания примеров заголовков из CSS-файла помогут понять, как с помощью задания полей, фоновых изображений и ширины с высотой можно эффективно управлять положением графики и текста:

```
#preamble h3 {
 background: transparent url(ch1.gif) no-
repeat top left;
 margin: 10px 0px 5px 0px;
 width: 388px;
 height: 44px;
}
#benefits h3 {
 background: transparent url(ch4.gif) no-
repeat top left;
 margin: 10px 0px 5px 0px;
 width: 388px;
 height: 44px;
}
#requirements h3 {
 background: transparent url(ch5.gif) no-
repeat top left;
 margin: 10px 0px 5px 0px;
 width: 388px;
 height: 44px;
}
```

Теперь заголовки удачно сочетаются с прочими элементами на странице (рис. 5).

Наконец, мы видим, как выглядят синие рамки:

```
#benefits {
 border-bottom: 1px dashed #8095AF;
}
#preamble {
 border-bottom: 1px dashed #8095AF;
}
```

Рамки добавляют зрительный контраст и одновременно выступают в роли разделителей между фрагментами текста (рис. 6).

А теперь хвостовик

Теперь, когда центральная часть страницы оформилась, можно заняться хвостовиком. Его содержимое просто: ссылки на инструменты контроля и фоновая картинка с головой коровы:

```
#footer {
 text-align: center;
 background: transparent url(bottom.gif) no-
repeat top left;
 height: 37px;
```

```
  margin: 0px -200px 0px -68px;
  padding-top: 88px;
}
#footer a:link, #footer a:visited {
  margin-right: 20px;
  }
```

Обратите внимание на отрицательные поля слева и справа: они помогают удержать хвостовик в границах объемлющего элемента (рис. 7).

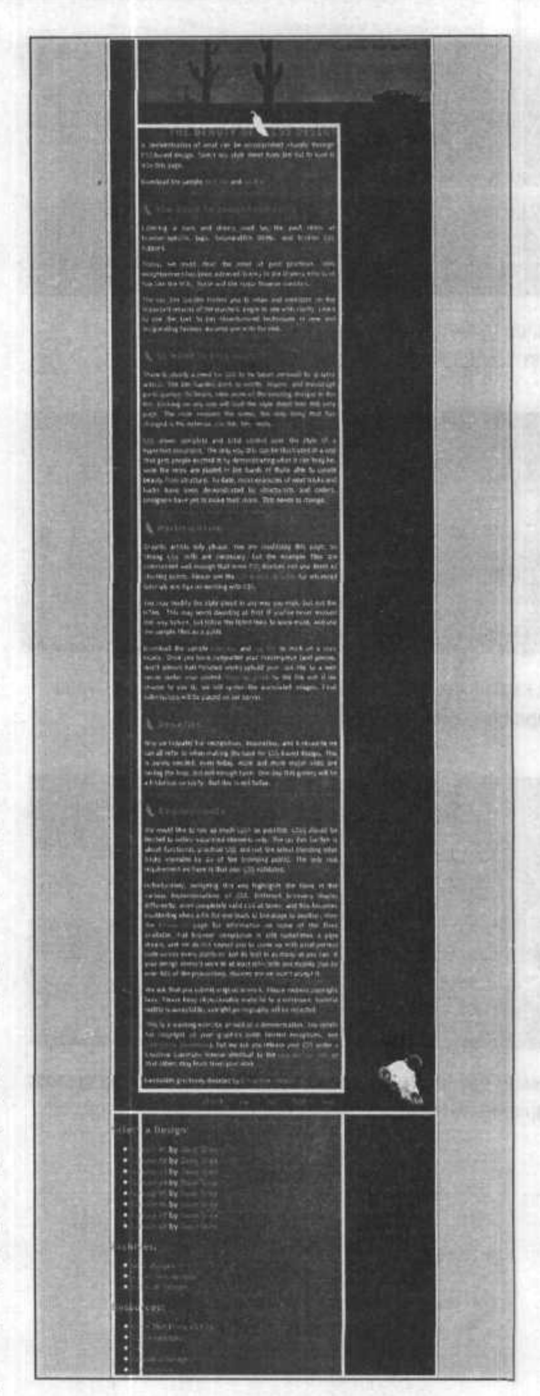


Рис. 7 ✦ После добавления хвостовика

Абсолютное позиционирование

Вы не забыли, что элемент `#container` позиционирован относительно? Так вот, список ссылок абсолютно позиционирован внутри относительно позиционированного `div`:

```
#linkList {
 position: absolute;
 top: 207px;
 right: 6px;
 width: 181px;
}
```

В результате список ссылок оказывается сбоку от основного текста. В отличие от относительного, при абсолютном позиционировании элемент размещается относительно своего родительского элемента[1]. Если такового не существует, то вычисляется позиция относительно корневого элемента `html`.

Чтобы наглядно увидеть, что происходит, достаточно отменить относительное позиционирование для элемента `#container` (рис. 8).

Природа абсолютного позиционирования такова, что элемент позиционируется относительно *позиционированного* объемлющего элемента или относительно элемента `html`, если объемлющего не существует. Когда с непосредственного родителя снято относительное позиционирование, элемент `#linkList` позиционируется относительно корневого элемента. После восстановления относительного позиционирования для элемента `#container` дизайн снова выглядит, как задумано.

[1] На самом деле, относительно ближайшего позиционированного (относительно или абсолютно) предка. – *Прим. пер.*

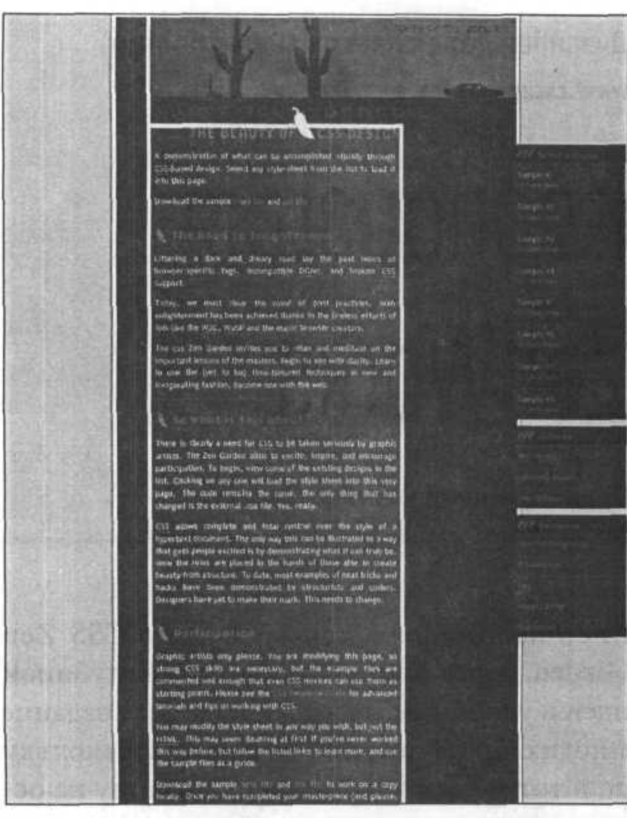


Рис. 8 ✦ Позиционированный элемент `#linkList` потерял связь с объемлющим его элементом `div #container`

СОВЕТ

Это свойство модели позиционирования полезно для верстки, поскольку позволяет определять точки отсчета для абсолютного позиционирования. Если ни один из предков не позиционирован, то абсолютная позиция отсчитывается от положения элемента `html`, то есть от краев окна браузера. В противном случае позиция отсчитывается от краев ближайшего позиционированного предка.

Как воспользоваться этой возможностью, если родитель не позиционирован? Задайте для родителя свойство `position:relative;`, не указывая ни `top`, ни `left`; на фактическом положении родителя это не отразится, зато установит новую точку отсчета. Подробнее на эту тему читайте статью Дугласа Боумена «Making the Absolute, Relative» на сайте Stopdesign (www.stopdesign.com/articles/absolute).

Закончим со ссылками

Поместив элемент `#linkList` в нужное место, можно применить к нему дополнительные стили. Прежде всего нужно задать поля и отступы равными 0:

```
#linkList ul {
 margin: 0px;
 padding: 0px;
}
```

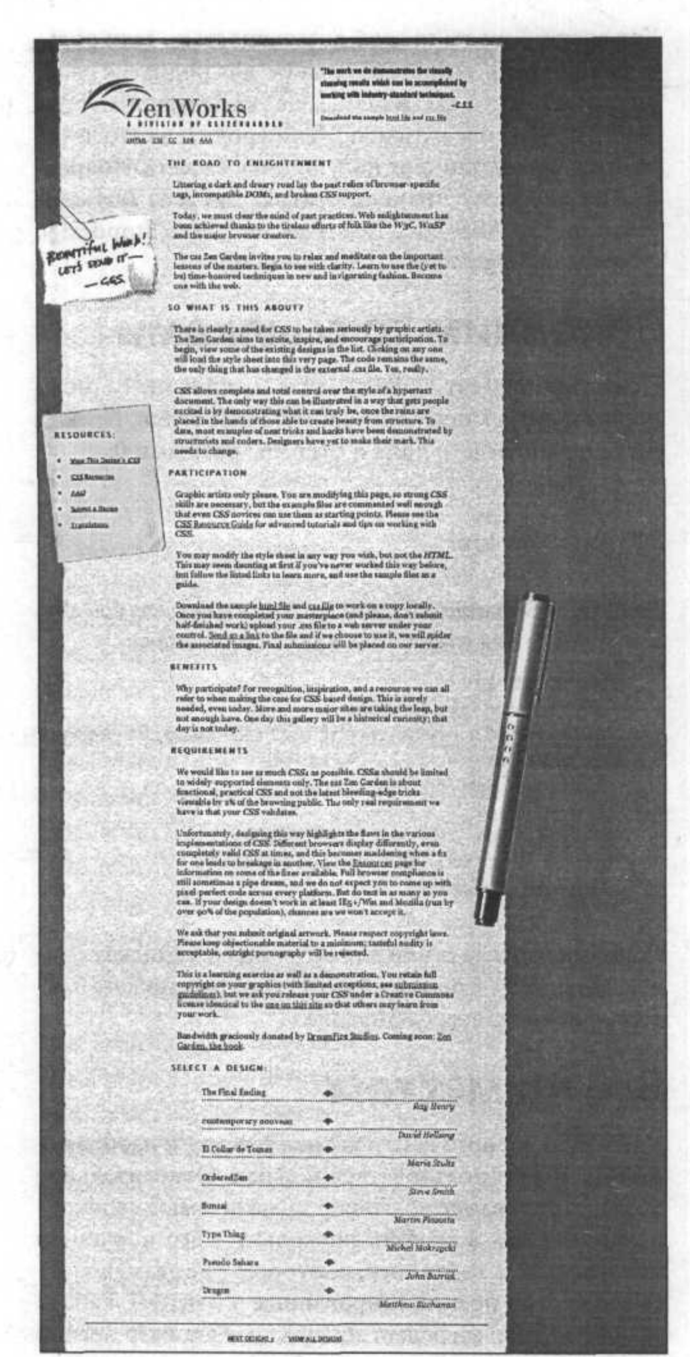
Далее в CSS-файле следуют другие стили для ссылок, в том числе для отдельных элементов списка:

```
#linkList li {
 list-style-type: none;
 background: transparent url(li-bg.gif) repeat
top center;
 padding: 8px 6px 8px 6px1
 margin: 0px;
 border-top: 1px solid #5D4831;
 border-bottom: 1px solid #221609;
 display: block;
}
```

Дополнительные стили для заголовков в списках ссылок, задания цветов, полей и отступов вы сможете найти в оригинальном CSS-файле.

Все по местам

Для эффективного контроля над текстом и изображениями можно пользоваться полями. Относительное позиционирование позволяет загнать самые непокорные элементы в стойло, поскольку с его помощью можно создать контейнер, в котором уже размещаются абсолютно позиционированные элементы. Работа «К югу от границы» дает прекрасный пример четкой темы и бесхитростного способа ее реализации с помощью CSS.



Дизайнер **Дерек Хансен** (Derek Hansen)
www.csszengarden.com/095

Корпорация ZenWorks

От конкуренции к согласию

Впервые открыв для себя сайт CSS Zen Garden, Дерек Хансен был поражен глубиной идеи и усилиями, затраченными на создание многих представленных работ. Поскольку он в тот момент переходил к дизайну на основе стандартов, то предпосылки к представлению собственного варианта дизайна уже существовали.

Хансену хотелось сделать что-то уникальное. Он тщательно исследовал все уже имеющиеся варианты. Больше всего ему понравились работы с четко выраженной темой, и он решил, что нужно создать нечто, содержащее в себе целый рассказ, но при этом отличное от всех прочих вариантов дизайна, представленных на сайте. В результате получилась «Корпорация ZenWorks».

Возьмем фирменный бланк

Как во всяком хорошо написанном документе, в «Корпорации ZenWorks» уделено много внимания удобной организации и простому представлению контента (рис. 1). Серебристый фон и простые шрифты задают общий тон:

```
body {
  background: #999 url(bk.jpg) repeat-x fixed;
  margin: 0;
  padding: 0;
  color: #33140E;
  font-family: Georgia, Times, serif;
  font-size: 80%;
}
h1,h2,h3,h4,h5,h6 {
  border: 0;
  margin: 0;
  padding: 0;
  font-family: "Lucida Grande", "Lucida Sans
Unicode", Verdana,
sans-serif;
  font-weight: normal;
}
h3 {
  text-transform: uppercase;
  font-size: 90%;
  word-spacing: 2px;
  letter-spacing: 2px;
  margin-left: -10px;
  font-weight: bold;
}
```

Обратите внимание на отрицательные поля у элементов h3 (рис. 2). Сейчас это выглядит немного странно, но, когда мы доберемся до конца, станет ясно, что это придает странице дополнительное очарование.

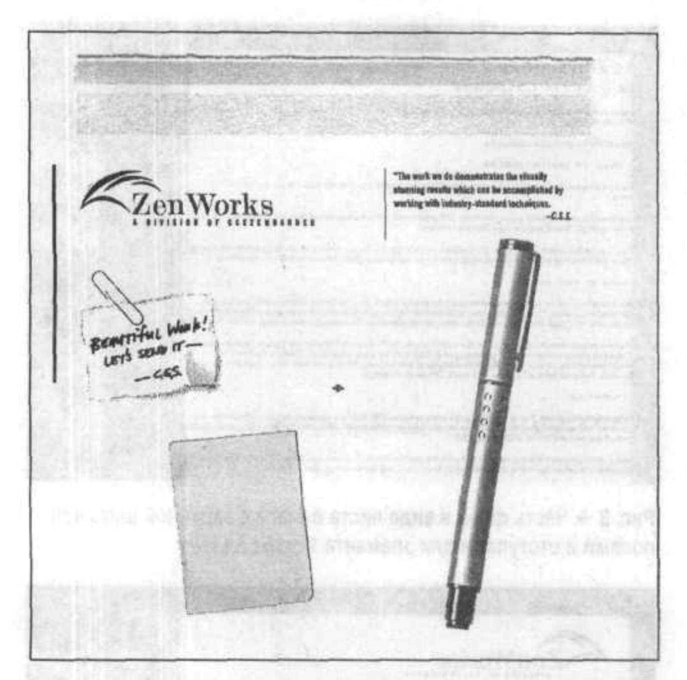


Рис. 1 ✦ Графика, использованная в «Корпорации ZenWorks»

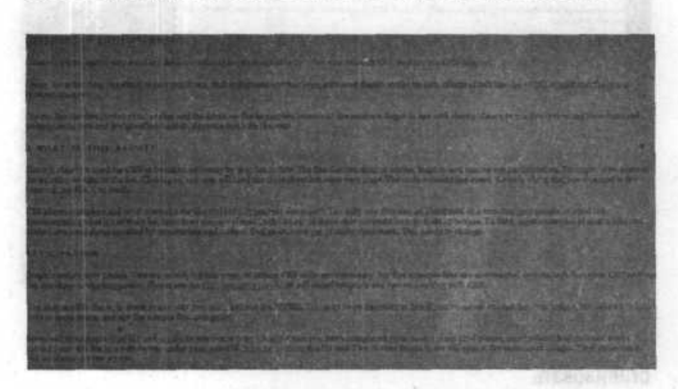


Рис. 2 ✦ Часть основного фона и шрифты в «Корпорации ZenWorks». Заголовки уходят за левый край вследствие отрицательных значений полей, а текст очень близко подступает к краю документа

ПРИМЕЧАНИЕ

Использование шрифта Lucida Grande для заголовков и Georgia для основного текста – это простое средство достижения элегантного результата. Хансен объяснил, что не хотел злоупотреблять типографией, главным образом потому, что старался уменьшить вес страницы и свести применение графики к минимуму. Из-за этого самоограничения результат походит на реальный корпоративный стиль письма, так что идея работает.

Бумагу, пожалуйста

Следующий шаг – поместить текст на бумагу и начать визуально организовывать слова. Для этого к разделу #container применяется следующий стиль:

```
#container {
  width: 750px;
  background: url(paper_mid.gif) repeat-y;
  margin: 50px 0 0 25px;
  padding-bottom: 20px;
}
```

В результате для контейнера задается фиксированная ширина 750 пикселей и добавляется основной графический образ листа бумаги, который повторяется вдоль вертикальной оси. Кроме того, для элемента `#container` устанавливаются поля и отступ вдоль нижнего края, чтобы оставить немного пустого места. Хотя из-за полей элементы h3 немного сдвинулись, они все еще находятся не там, где нужно (рис. 3), но вскоре мы это исправим.

Рис. 3 ✦ Часть фона в виде листа бумаги с заданной шириной, полями и отступами для элемента `#container`

Фирменный логотип

Поскольку в основу дизайна положен образ фирменного бланка, необходимо добавить фирменный логотип и соответствующие картинки. Вот как логотип присоединяется к элементу `h1`:

```
#pageHeader {
 height: 100px;
 padding: 20px 0 20px 50px;
}
#pageHeader h1 {
 height: 90px;
 width: 340px;
 background: url(zen_logo.gif) no-repeat;
}
```

Следующее правило скрывает текст логотипа, оставляя картинку:

```
#pageHeader h2 span, #pageHeader h1 span {
 display: none;
}
```

Теперь самое время установить поля для заголовков и текста:

```
#linkList, #supportingText, #preamble {
 margin: 10px 100px 10px 200px;
}
```

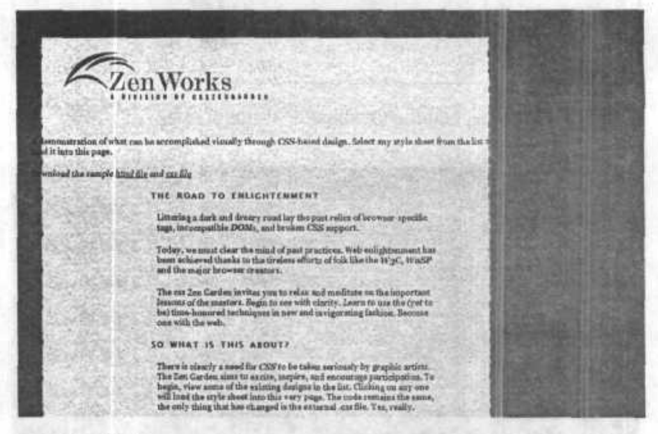


Рис. 4 ✦ Добавление фирменного логотипа. Пока что видно содержимое раздела `#quickSummary`, его еще предстоит стилизовать

Логотип оказался там, где и ему и полагается быть, в левом верхнем углу листа (рис. 4). Помимо логотипа, на странице присутствует еще и цитата, не без юмора подписанная инициалами C.S.S. Достигается это путем применения к элементу `#quickSummary` правила, которое позиционирует цитату, скрывает находящийся под ней текст и стилизует текст внутри раздела:

```
#quickSummary {
 position: absolute;
 top: 50px;
 left: 450px;
 background: url(subtext.gif) no-repeat;
 height: 100px;
 width: 280px;
}
```



Рис. 5 ✦ Страница стала больше похожа на фирменный бланк



Рис. 6 ✦ Добавление верхнего края листа бумаги



Рис. 7 ✦ Использование раздела #footer для стилизации верхней части страницы

ПРИМЕЧАНИЕ

Если бы вы специально разрабатывали XHTML-разметку под этот дизайн, то для раздела `#footer` следовало бы придумать какое-то другое имя, более соответствующее его расположению. То, что хвостовик вынесен в верхнюю часть страницы, обусловлено природой сайта CSS Zen Garden. Зато мы наглядно продемонстрировали, что любой элемент можно абсолютно позиционировать в произвольном месте страницы.

```
#quickSummary p.p1 {
 display: none;
}
#quickSummary p.p2 {
 padding: 78px 0 0 15px;
 font-size: 10px;
 font-family: «Lucida Grande», «Lucida Sans
Unicode», Arial,
sans-serif;
}
```

Сочетание графики и стилизованного текста создает впечатление традиционного фирменного бланка (рис. 5). Но верхний край бумаги, пожалуй, излишне ровный. С помощью дополнительных `div`, присутствующих в XHTML-разметке, Хансен добавил фоновое изображение, исправляющее этот недостаток:

```
#extraDiv1 {
 position: absolute;
 top: 20px;
 left: 23px;
 width: 750px;
 height: 50px;
 background: url(paper_top.gif) no-repeat;
}
```

Заметьте, что за счет абсолютного позиционирования удалось точно совместить верхний край листа бумаги с остальным содержимым страницы (рис. 6). Чтобы покончить с приданием корпоративного облика странице, применим к элементу `#footer` стиль, который позиционирует его в верхней части, сразу под логотипом:

```
#footer {
 position: absolute;
 top: 160px;
 left: 150px;
 font-size: 80%;
 border-top: 1px solid #6E512F;
 padding-top: 3px;
 width: 575px;
 text-transform: uppercase;
 text-decoration: none;
 font-family: «Lucida Grande», «Lucida Sans
Unicode», Arial,
sans-serif;
}
```

Это дает нам список ссылок и сплошную разделительную линию, после чего шапку бланка можно считать готовой (рис. 7).

Размещение ссылок

Хансен решил, что у каждого набора ссылок должен быть свой визуальный облик. Поля для элемента #linkList уже установлены, а стили для раздела Select a Design

вы найдете в оригинальном CSS-файле – поищите там селекторы #lselect и #larchives. В результате получается симпатичный список ссылок и область навигации по архивам, расположенная внизу страницы (рис. 8).

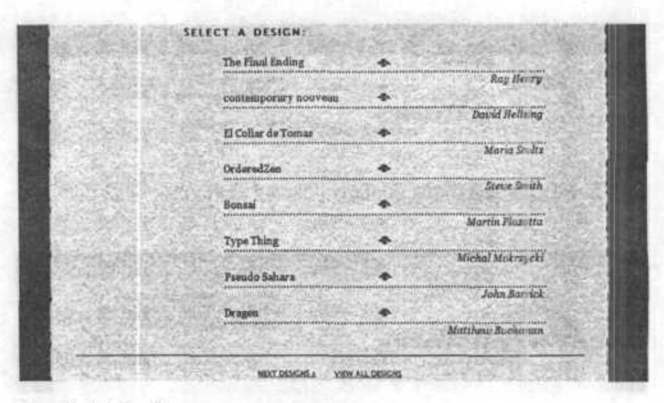


Рис. 8 ✦ Дизайн ссылок на архивы

Немного индивидуальности

Кто сказал, что корпоративная тема должна быть подобна пиджаку, застегнутому на все пуговицы? Хансен добавил ряд неформальных черточек. Во-первых, с листом бумаги скреплена записка, для этой цели задействован селектор `#pageHeader h2`:

```
#pageHeader h2 {
 background: url(paper_clip.gif) no-repeat;
 width: 320px;
 height: 262px;
 position: absolute;
 top: 250px;
 left: -6px;
 z-index: 5;
}
```

Как и раньше, с помощью абсолютного позиционирования Хансен сумел поместить изображение точно туда, куда хотел. Под запиской «наклеен» листочек, на котором разместились ссылки на ресурсы. Стили для ссылок посмотрите сами в исходном CSS-файле, а вот способ позиционирования листочка мы покажем здесь:

```
#lresources {
 position: absolute;
 top: 600px;
 left: 10px;
 width: 200px;
 height: 257px;
 background: url(sticky.gif) no-repeat;
 font-family: «Lucida Grande», «Lucida Sans
Unicode», Arial,
sans-serif;
 color: #000;
}
```

Теперь в левой части страницы есть два абсолютно позиционированных элемента, придающих ей более реалистичный вид (рис. 9). Последний визуальный элемент – для полноты картины – это ручка. Чтобы поместить ее в нужное место, Хансен воспользовался селектором `#extraDiv2` и, конечно же, абсолютным позиционированием (рис. 10).

```
#extraDiv2 {
 position: absolute;
 top: 900px;
 left: 700px;
 background: url(pen.gif) no-repeat;
 width: 140px;
 height: 547px;
}
```

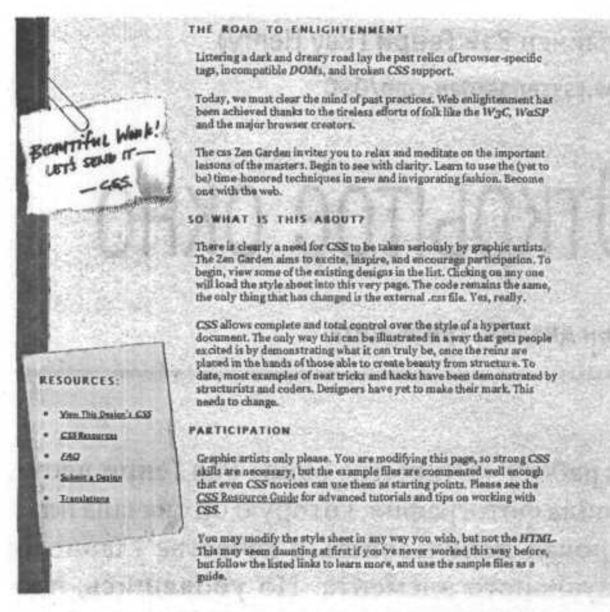
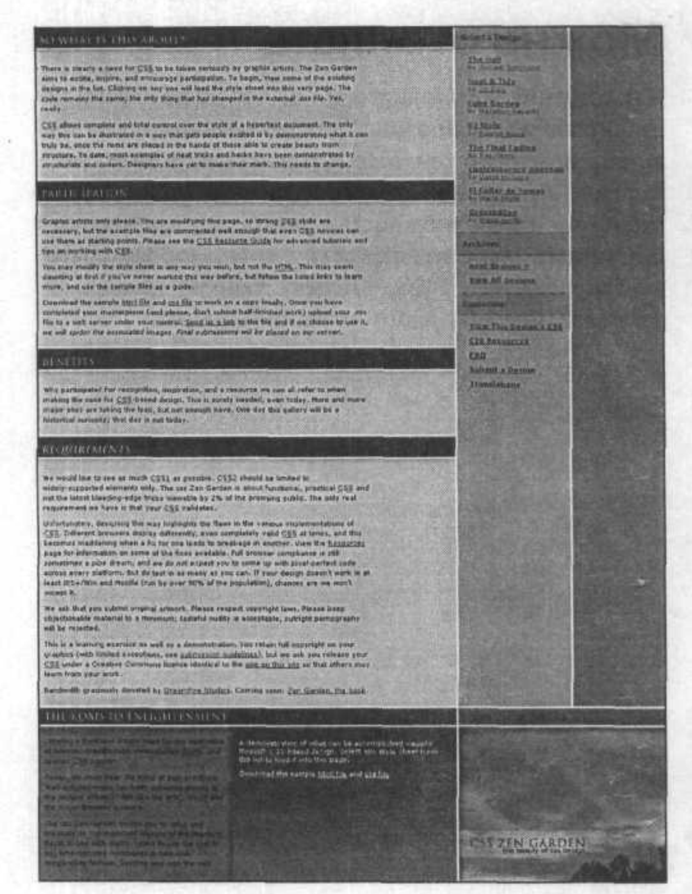


Рис. 9 ✦ Изображения приколотой скрепкой записки и приклеенного листочка абсолютно позиционированы

Рис. 10 ✦ Ручка дополняет картину и не дает корпоративному имиджу стать уж слишком серьезным

Можно отсылать

Для создания уникального, ни на что не похожего дизайна работа «Корпорация ZenWorks» опирается на четко определенную тему. Несмотря на некоторые сложности – к примеру, ограничения в использовании шрифтов и невозможность с помощью прозрачного GIF создать такие же эффекты, как с помощью альфа-канала в формате PNG, – Дереку Хансену удалось создать элегантный и запоминающийся образ.



Дизайнер **Рэй Генри** (Ray Henry)
www.csszengarden.com/090

Открытое окно

Один дизайн, три макета

На работу «Открытое окно» Рэя Генри вдохновила фотография, которую он сначала планировал использовать в качестве главного визуального элемента. Но убоявшись, что результат будет слишком предсказуем, он решил поискать новые подходы. Ему приходилось видеть верстку с фиксированными элементами для навигации и баннеров, но никогда не встречался дизайн, где бы наиболее существенные элементы размещались в нижней, а не верхней части страницы.

Так как Internet Explorer для Windows не поддерживает фиксированного позиционирования, то Генри понимал, что ему придется приложить дополнительные усилия для того, чтобы дизайн выглядел приемлемо в разных браузерах. Поэтому он прибег к различным трюкам, чтобы во всех современных браузерах был виден тот или иной аспект «Открытого окна».

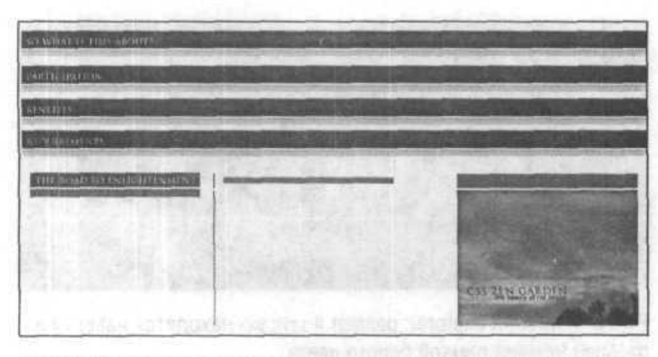


Рис. 1 ✦ Графика, использованная в «Открытом окне», включая и фотографию, послужившую источником вдохновения

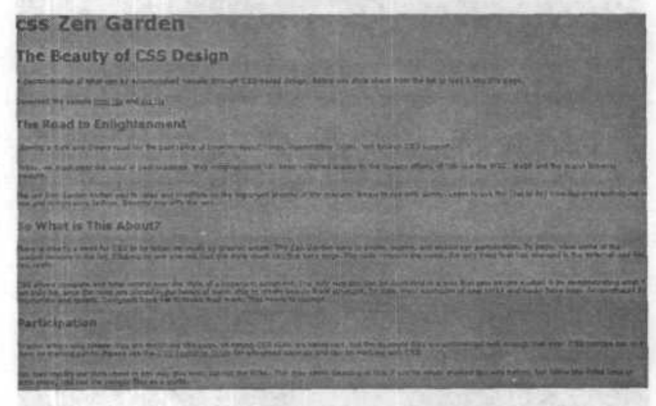


Рис. 2 ✦ Основа: страница выглядит одинаково во всех браузерах

Фундамент

Начав с элементов body и container, Генри отключил выставляемые по умолчанию поля и отступы и задал шрифты:

```
body#css-zen-garden {
  background: #A1B4CD;
  margin: 0;
  padding: 0;
  font-family: Verdana, arial, sans-serif;
  font-size: 11px;
}
```

Далее он задал для блока #container ширину 100%, выбрав тем самым текучую верстку, при которой страница заполняет всю площадь окна:

```
#container {
  width: 100%;
  text-align: left;
 }
```

В данный момент верстка выглядит одинаково во всех браузерах (рис. 2).

Стилизация раздела Intro

После позиционирования элемента #intro дизайн становится намного интереснее:

```
#intro {
 width: 100%;
 height: 276px;
}
html>body div#intro {
 position: fixed;
 bottom: 0px;
 right: 0;
}
html*#intro {
 border-bottom: 3px solid #fff;
 position: fixed;
 bottom: -10px;
}
#intro {
 height: 286px;
 voice-family: "\"}\"";
 voice-family: inherit;
 height: auto;
}
```

В этих правилах вы найдете целый ряд приемов, призванных «подсунуть» каждому браузеру свой стиль. Первый селектор #intro распознается всеми браузерами. Следующие два правила предназначены только для браузеров, поддерживающих фиксированное позиционирование, при этом третье содержит описание рамки, которое Internet Explorer 6 для Windows интерпретирует правильно. Наконец, вы видите трюк на основе модели прямоугольных областей (Box Model Hack), которым Генри воспользовался, чтобы компенсировать ошибку при вычислении высоты в правиле #intro, характерную для Internet Explorer 5.

Теперь в Internet Explorer раздел #intro находится в верхней части страницы (рис. 3) и имеет белую нижнюю рамку. В то же время в браузерах Mozilla и Safari этот раздел фиксирован и расположен внизу (рис. 4).

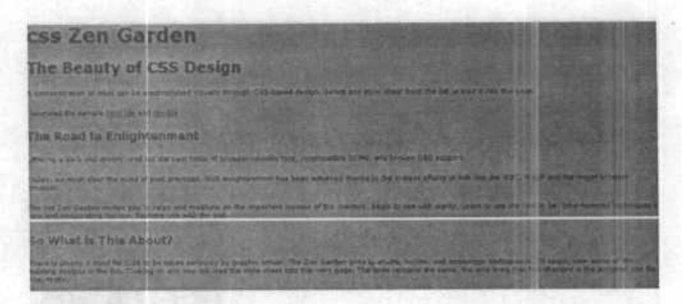


Рис. 3 ✦ Internet Explorer: раздел #intro находится наверху и отделен нижней рамкой белого цвета

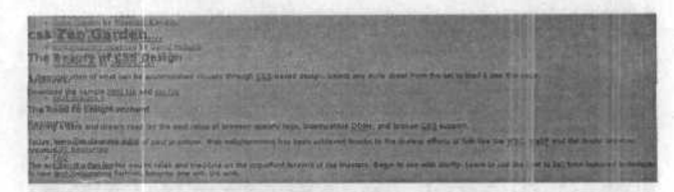


Рис. 4 ✦ Mozilla (и Safari): раздел #intro фиксирован в нижней части страницы и перекрывает расположенный в том же месте текст

Вверх, вниз, вправо, влево

На данный момент верхняя и нижняя части страницы определены. На самом деле это, конечно, один и тот же раздел. Продолжая применять фильтрацию, Генри добавил стили, которые отображают структурные элементы в нижней или верхней части в зависимости от браузера.

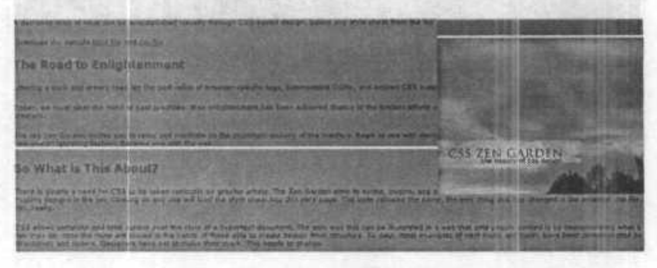


Рис. 5 ✦ В Internet Explorer фотография оказывается в правом верхнем углу страницы

Шаг вправо...

Сначала добавляется фотография, которая определяет цветовую гамму:

```
#pageHeader {
 background: url(pageHead_bg.jpg) no-repeat
top left;
 width: 332px;
 height: 283px;
 position: absolute;
 right: 0;
 top: 0;
}
```

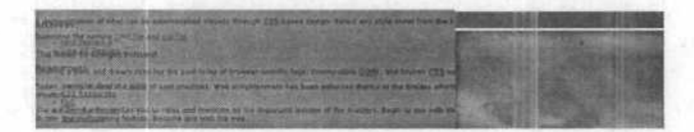


Рис. 6 ✦ В Mozilla фотография оказывается в правом нижнем углу страницы, и пока мы ее полностью не видим

Это правило позиционирует раздел #pageHeader и в Internet Explorer (рис. 5), и в Mozilla, хотя, как видите, положение определяется относительно того места, куда браузер поместил фиксированный раздел (рис. 6.). Интересно, что в браузере Opera картинка пока не видна, с этим еще предстоит повозиться.

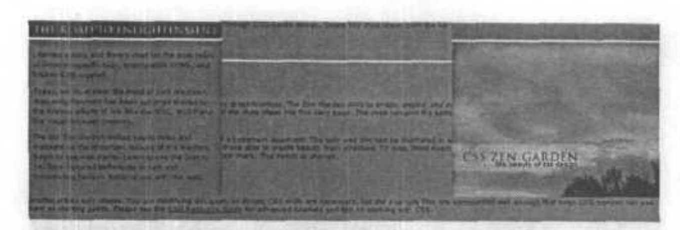


Рис. 7 ✦ В Internet Explorer раздел #preamble позиционирован в левом верхнем углу

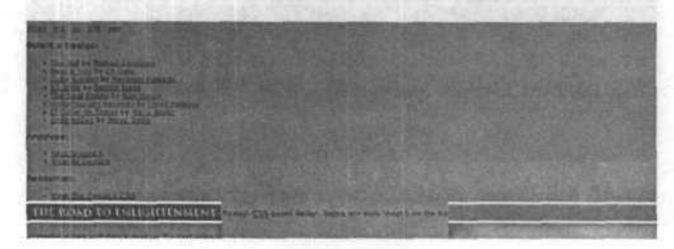


Рис. 8 ✦ В Mozilla (и Safari) раздел #preamble, как и #pageHeader, еще не виден полностью

> **ПРИМЕЧАНИЕ**
>
> Подробнее о трюке Box Model Hack читайте на странице http://tantek.com/CSS/Examples/boxmodelhack.html.

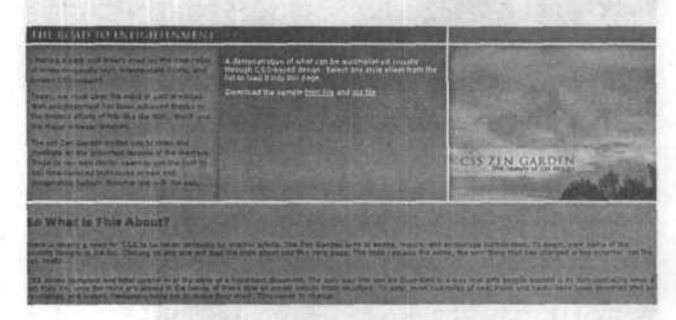


Рис. 9 ✦ В Internet Explorer раздел оказался вверху

Прыжок влево...

Следующие правила CSS задают стиль для раздела #preamble, правда, из соображений экономии места мы не приводим некоторые правила для внутренних элементов:

```
#preamble {
 background: #A1B4CD url(preamble_bg.gif)
repeat-y top right;
 height: 283px;
 width: 312px;
 margin: 0;
 padding: 0;
 position: absolute;
 left: 0;
 top: 0;
}
html>body #preamble {
 position: absolute;
 left: 0;
}
```

Снова вы видите, как селектор #preamble задает способ позиционирования блока (рис. 7), причем для браузеров, поддерживающих селекторы дочерних элементов, этот блок в следующем правиле прижимается к левому краю (рис. 8).

А теперь в середину

Раздел #quickSummary Генри поместил точно посередине между #pageHeader и #preamble:

```
#quickSummary {
 background: url(quickSum_bg.gif) repeat-x top
right;
 border-right: 3px solid #fff;
 margin: 0 332px 0 312px;
 padding: 52px 0 0 0;
 height: 283px;
 voice-family: "\"}\"";
 voice-family: inherit;
 height: 231px;
}
```

Единственный трюк здесь – это Box Model Hack в разделе #quickSummary. Он нужен, чтобы исправить ошибку при вычислении высоты в браузерах, где некорректно реализована модель прямоугольных областей.

В этот момент все три варианта верстки более-менее видимы. В Internet Explorer раздел отображается целиком вдоль верхнего края (рис. 9), а в Mozilla он виден частично и находится внизу (рис. 10). Кстати, кое-что начало уже проявляться и в браузере Opera (рис. 11).

Стилизация контента и ссылок

Следующий шаг – применить стили к контенту и ссылкам. Все правила можно посмотреть в исходном CSS-файле на сайте Zen Garden, но некоторые наиболее интересные заслуживают упоминания здесь. Прежде всего, обратите внимание на то, как с помощью свойства `float` контент перемещен в левую часть страницы:

```
#supportingText {
 width: 100%;
 float: left;
 margin-right: -332px;
 margin-top: -3px;
}
```

Интересно отметить, что даже при ширине 100% отрицательные поля позволяют управлять внешним видом текстового контента. Отрицательное значение `margin-right` дает тот же эффект, что положительное значение `padding-right`, и гарантирует, что текст не выйдет за пределы области, ограниченной левым краем фотографии. Однако в случае, когда ширина равна 100%, использовать свойство `padding-right` нельзя, поскольку тогда ширина элемента `#supportingText` оказалась бы равна `100% + 332px`. К разделу `#footer` применяются стили, учитывающие различия между браузерами:

```
#footer {
 background: #d6e0ee;
 margin: 0 332px 0 0;
 border-right: 3px solid #fff;
 border-top: 3px solid #fff;
 border-bottom: 3px solid #fff;
 padding: 15px 0;
 text-align: center;
}
html>body #footer {
 background: transparent;

 position: fixed;
 right: 0;
 bottom: 261px;
 padding: 0;
 margin: 0 160px 0 0;
 border-right: 0;
 border-top: 0;
 border-bottom: 0;
 text-align: left;
}
```

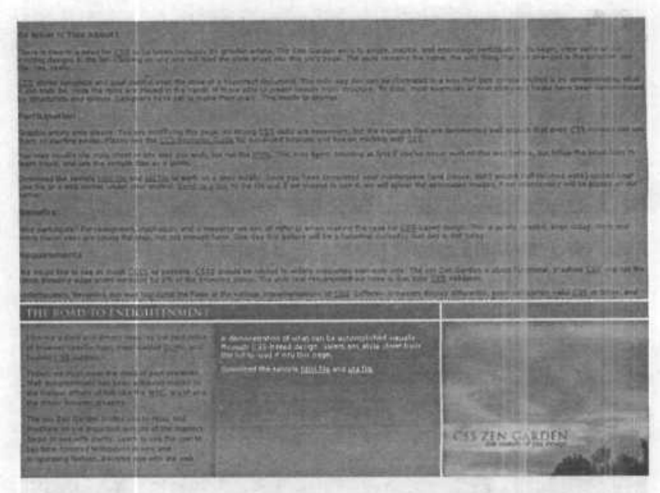


Рис. 10 ✦ В Mozilla раздел прижат к нижнему краю



Рис. 11 ✦ В Opera пока отображается только центральная часть

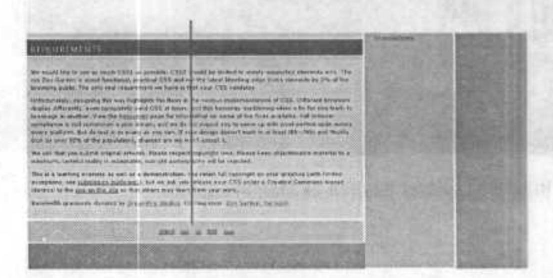


Рис. 12 ✦ В Internet Explorer раздел `#footer` отображается под контентом

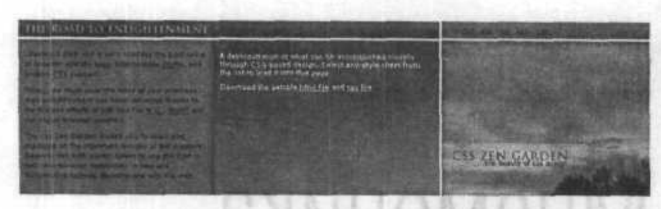


Рис. 13 ✦ В Mozilla раздел `#footer` отображается в фиксированной нижней части страницы

ПРИМЕЧАНИЕ

Подробнее о технике применения спецификатора @media в браузере Opera 7 см. страницу www.virtuelvis.com/archives/145.html.

В результате хвостовик оказывается в верхней видимой части страницы для Internet Explorer (рис. 12) и в нижней части для Mozilla (рис. 13).

А теперь займемся браузером Opera

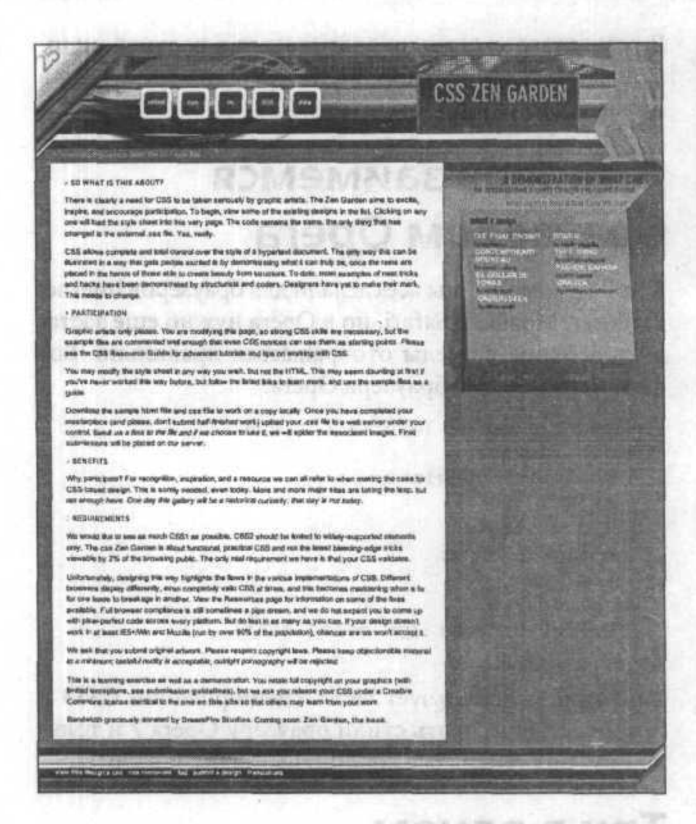
К этому моменту мы все сделали для браузеров Internet Explorer, Mozilla и Safari, но в Opera нужно еще заставить верхние разделы отображаться правильно. Вот фрагмент CSS для браузера Opera:

```
@media all and (min-width: 0px) {
div#intro {
position: absolute;
right: 0;
width: 100%;
height: 283px;
top: 0;
z-index: 0;
border-bottom: 3px solid #fff;
}
```

Этот прием использует часть спецификации CSS3 и позволяет адресовать стили браузеру Opera 7 и только ему. Вот теперь все хорошо выглядит и в Opera.

Три в одном

Оригинальная идея работает как задумано в браузерах Mozilla и Safari; для Internet Explorer найден обходной путь, позволяющий приемлемо отобразить страницу, а для Opera пришлось пойти на компромисс – фотография находится вверху, но, в отличие от Internet Explorer, она фиксирована, так что страница прокручивается под ней. Своим названием работа «Открытое окно» обязана соотношению между верхом и низом, найденному в результате экспериментов Генри.



Дизайнер **Дэйв Ши** (Dave Shea)
www.csszengarden.com/025

Мнемоника

Дизайнерские трюки, обеспечивающие текучую верстку и дополнительные эффекты

Текстура «Мнемоники» появилась в результате смешения органических форм со строгими линиями и геометрическими фигурами. Тонкие усики пересекают толстые горизонтальные линии в заголовке, а резкая цветовая гамма придает неповторимость дизайну.

В этой работе Ши хотел создать что-то совершенно новое. В результате получилась уникальная верстка, которой, возможно, недостает внутренней красоты, зато за счет визуальной целостности и внимания к деталям возникает впечатление оригинальной, хорошо продуманной концепции.

Стремлением к оригинальности продиктованы многие решения, принятые как на этапе проектирования, так и во время кодирования. Мы рекомендуем внимательно прочитать хорошо прокомментированный CSS-файл, находящийся на сайте Zen Garden (www.csszengarden.com/025/025-comments.css). Отдельные изображения, использованные в «Мнемонике», приведены на рис. 1 ниже.

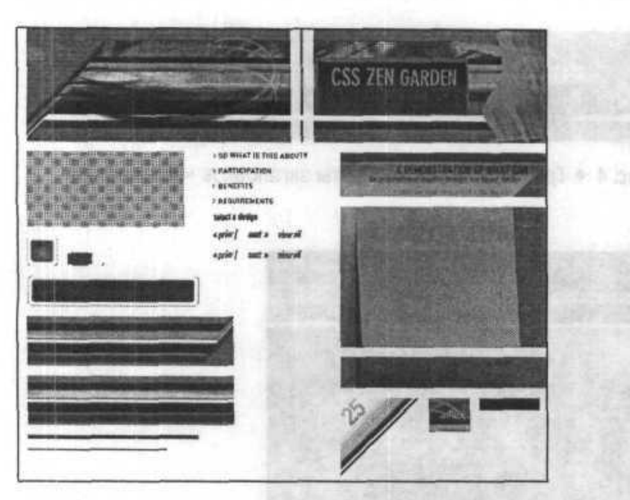


Рис. 1 ✦ Все изображения, использованные в «Мнемонике»

Рис. 2 ✦ Вид страницы после применения начальных стилей, задающих цвета ссылок и шрифты

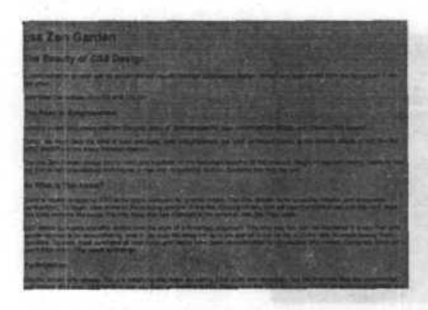


Рис. 3 ✦ Первая фоновая картинка присоединена к элементу body

> **ПРИМЕЧАНИЕ**
>
> Цветной вариант рис. 1–3 см. на вкладке.

Решение задач

После применения начальных стилей (они здесь не приводятся, но результат показан на рис. 2) возникла первая проблема. Для «Мнемоники» нужны были две желтых картинки для фона элемента body: одна повторяющаяся по вертикали и вторая – перекрывающая ее и повторяющаяся по горизонтали.

```
body {
  font-size: small;
  font-family: arial, sans-serif;
  background: #B0A40B url(bg-1.gif) top left
repeat-x;
  margin: 0px;
  padding: 0px;
}
```

Но к элементу body присоединена горизонтально повторяющаяся фоновая картинка (рис. 3), а CSS не позволяет присоединять к одному элементу несколько изображений. Так что оставалось лишь использовать два элемента. К счастью, элемент #container находится внутри body и заключает в себе всю остальную разметку, поэтому вторую картинку можно присоединить к нему:

```
#container {
 background: transparent url(bg-2.gif) top
right repeat-y;
}
```

Это работает. Поскольку графика в этом дизайне сосредоточена в основном в заголовке, то дальше нужно было придумать, как интегрировать каждую картинку в расширяющуюся полоску, идущую поперек страницы, и при этом сохранить единство зрительного образа.

Организация графики

Взглянув на окончательный вид страницы, вы заметите много отдельных визуальных элементов. Надо ли расположить некоторые поверх других, воспользовавшись абсолютным позиционированием? Или лучше сохранить все вместе в виде одного большого изображения?

Решение находится посередине. Выбор того, какие элементы объединяются в один JPEG-файл, а какие выделяются в отдельные GIF-файлы, зависит от личных предпочтений дизайнера и ограничений, налагаемых разметкой. Не существует «единственно верного» ответа, два дизайнера могут прийти к различным решениям.

Поскольку изменять HTML-разметку сайта Zen Garden нельзя, то было принято решение присоединить изображения к двум заголовкам внутри раздела #pageHeader и, быть может, задействовать сам этот div. Таким образом, мы можем взять максимум три картинки, поэтому левая часть страницы (за вычетом левого верхнего графического уголка) была сохранена в виде одного изображения, а правая часть – в виде другого. Третий – узкий – GIF повторяется горизонтально под каждой стороной.

Не сразу было понятно, как можно заставить такую графику расширяться в случае текучей верстки; казалось, что фиксированные размеры больше подходят для верстки с фиксированной шириной. Однако между двумя сторонами заголовка есть разрыв, который можно было бы заполнить однородными полосками, допускающими расширение (рис. 4). Это сделано специально, так как Ши заранее решил, что верстка должна быть именно текучей.

Итак, картинки выбраны, посмотрим, как их лучше скомбинировать. Повторяющееся изображение должно быть размещено под всеми остальными (рис. 5):

```
#pageHeader {
 height: 171px;
 padding: 0px;
 background: transparent url(bg-4.gif) top
left repeat-x;
}
```

Неизменяемость HTML-разметки Zen Garden сужает диапазон доступных возможностей, но все же их остается достаточно для того, чтобы добиться нужного результата за счет комбинирования графических элементов. Поскольку текст элементов h1 и h2 внутри раздела #pageHeader отображается на фоне одной и той же картинки (правой стороны заголовка), то применить стиль нужно только к одному из этих двух заголовков, и тогда будет виден весь текст. Таким образом, второй заголовок можно использовать для дальнейшей подстановки изображения, и мы присоединим к нему картинку, которая появится слева (рис. 6 и 7):

```
#pageHeader h1 {
 padding: 0px;
 margin: 0px;
 float: left;
 width: 396px;
 height: 171px;
 background: transparent url(cr-tl.jpg) top
right no-repeat;
}
#pageHeader h1 span {
 display: none;
}
```



Рис. 4 ✦ Графические компоненты заголовков «Мнемоники»

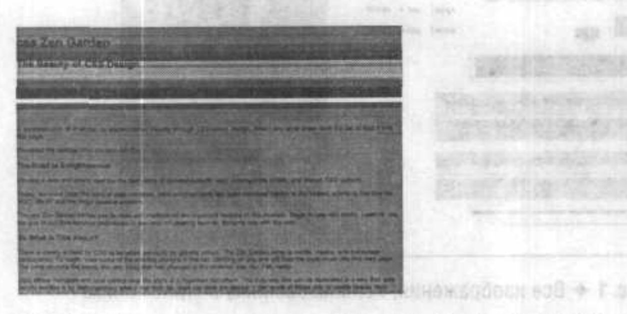


Рис. 5 ✦ Первым размещается повторяющееся изображение, являющееся частью заголовка

ПРИМЕЧАНИЕ

На этапе первоначального проектирования нужно заранее рассмотреть подобные сценарии. Если о верстке начать думать только на этапе составления таблиц стилей, то не исключено, что для обеспечения приемлемого поведения при изменении размера всю графику придется радикально переделать.

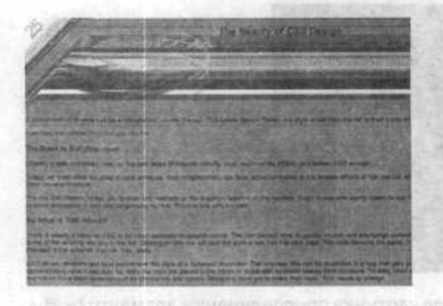


Рис. 6 ✦ Свободный элемент, к которому можно присоединить вторую картинку, образующую заголовок...



Рис. 7 ✦ ... если оба текстовых элемента помещены на одну картинку



Рис. 8 ✦ Версия «Мнемоники» без специальных эффектов. Так выглядит страница в Microsoft Internet Explorer

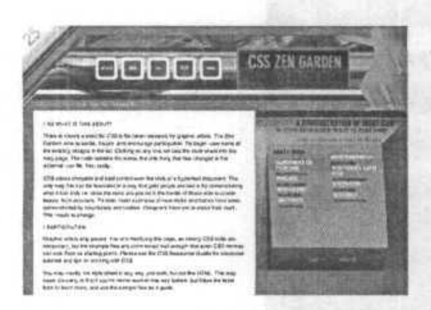


Рис. 9 ✦ А так выглядит «Мнемоника» в браузерах, поддерживающих специальные эффекты

После того как левая часть заголовка построена, его текст отображается поверх картинки в правой части, которая просто присоединяется к свободному элементу h2:

```
#pageHeader h2 {
 padding: 0px;
 margin: 0px;
 position: absolute;
 top: 0px;
 right: 0px;
 width: 350px;
 height: 171px;
 background: transparent url(cr-tr.jpg) top
right no-repeat;
}
#pageHeader h2 span {
 display: none;
}
```

Идем дальше

Мы пропустим описание значительной части процесса верстки страницы и расскажем о некоторых нетривиальных эффектах в «Мнемонике». В конце оригинального CSS-файла есть место, сопровождаемое комментарием о том, что начинаются специальные эффекты. Если исключить из файла все, начиная с этого места, то получится страница, изображенная на рис. 8. Собственно, именно так и будет выглядеть окончательный результат в Internet Explorer. Однако в браузерах Firefox или Safari страница будет выглядеть так, как показано на рис. 9.

Суть проблемы

Выбрать уникальные элементы на странице, которым не назначен ни класс, ни идентификатор, можно с помощью имеющихся в CSS2 селекторов. Беда в том, что не все браузеры их поддерживают. Ши воспользовался этими селекторами, но вообще-то добиться нужного результата можно было бы и проще: достаточно изменить разметку, чтобы на нужные элементы можно было сослаться.

Когда составляется разметка, нужно стараться поддерживать баланс между простотой кода и возможностью выбрать затем необходимые элементы. При создании сайта Zen Garden ставилась задача добиться максимальной гибкости, но в некоторых случаях ее все же оказалось недостаточно. В большинстве проектов HTML-разметка и CSS разрабатываются параллельно, так что вы можете поступать так, как удобно.

Список вариантов дизайна

В Internet Explorer список вариантов дизайна представлен в виде одной вертикальной колонки (рис. 10), а в Firefox и Safari - в виде двух колонок, наклоненных под тем же углом, что левый край фона (рис. 11). Чтобы достичь такого эффекта, нужно выбрать и позиционировать отдельные элементы списка. Для этого применяются селекторы дочерних и сестринских элементов, которые определены в спецификации CSS2, но не поддерживаются браузером Internet Explorer для Windows:

```
#lselect ul>li {
 position: relative; left: -14px;
}
#lselect ul>li+li {
 position: relative; left: -12px;
}
#lselect ul>li+li+li {
 position: relative; left: -10px;
}
...
```

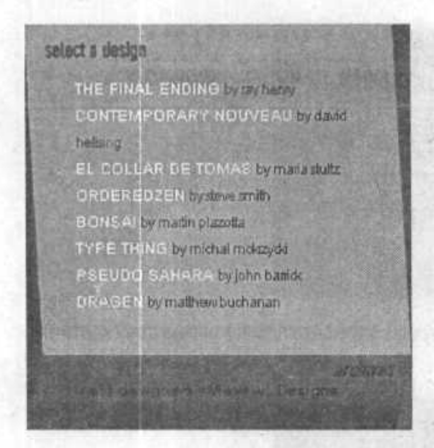


Рис. 10 ✦ Список вариантов дизайна в Internet Explorer

ПРИМЕЧАНИЕ

Многоточие означает, что это не конец кода.

Первый селектор выбирает любой элемент `li`, являющийся непосредственным потомком элемента `ul`, находящегося внутри раздела `#lselect`, в данном случае – все восемь таких элементов. Второй селектор аналогичен, только элементу `li` должен предшествовать еще хотя бы один такой же элемент, а следовательно, выбираются все элементы `li`, кроме первого. Третий селектор делает то же самое, но выбираемому элементу должно предшествовать уже не менее двух таких же элементов, и так далее. Строя каскадный список правил, каждое из которых переопределяет предыдущее, мы в конечном итоге позиционируем все восемь ссылок по отдельности. Это излишне запутанное решение простой, в сущности, проблемы; как правило, вы можете изменить разметку, добавив дополнительные классы, и тем самым избежать написания длинных селекторов CSS. Именно так мы и рекомендуем поступать.



Рис. 11 ✦ Список вариантов дизайна в браузерах того же класса, что Firefox и Safari

Ссылки на следующий и предыдущий вариант дизайна

Последнее, что осталось сделать, – это заменить невыразительные ссылки под списком вариантов дизайна более симпатичными графическими эквивалентами (рис. 12 и 13). Как и раньше, сложность состоит в том, что для этих элементов не определены отдельные классы, так что их не удается выбрать простым



Рис. 12 ✦ Вид ссылок Next designs и View All Designs в Internet Explorer для Windows



Рис. 13 ✦ Вид ссылок Prior, Next и View All в браузерах типа Firefox и Safari

способом. Но в спецификации CSS2 описаны не только селекторы дочерних и сестринских элементов, есть и другие. В данном случае удобно воспользоваться селекторами атрибутов, поскольку они позволяют выбрать элемент в зависимости от его уникальных характеристик. Для всех ссылок в HTML-разметке определены различные клавиши доступа:

```
<a href="http://www.mezzoblue.com/zengarden/
alldesigns/" title="View every submission to
the Zen Garden. Accesskey: w" accesskey="w">
```

Такого рода уникальность идеально подходит для применения селекторов атрибутов:

```
#larchives ul>li+li a[accesskey="w"] {
  display: block;
  width: 35px;
  height: 0;
  overflow: hidden;
  padding-top: 18px;
  background: transparent url(archive-b3a.gif)
 top left no-repeat;
  position: absolute;
  top: 16px;
  left: 170px;
  color: #958500;
 }
```

Как следует из самого названия, селектор атрибутов позволяет выбрать элемент, для которого значение именованного атрибута соответствует указанному. В листинге выше выделена наиболее интересная часть селектора: поскольку у нужного нам элемента действительно есть атрибут accesskey и его значение равно w, то имеет место соответствие, так что к элементу применяется стилистическое правило.

Возможно, вы задаетесь вопросом, почему для списка вариантов дизайна мы применили такие сложные селекторы дочерних элементов, ведь у них тоже имеются атрибуты accesskey. Вообще-то вы правы, имело смысл так и поступить.

Как сделать правильный выбор

При разработке «Мнемоники» были различные варианты. Ши мог бы по-другому разбить изображение заголовка в Photoshop, а мог бы совершенно иначе подойти к выбору ссылок, входящих в список вариантов дизайна. Если вы понимаете, какие имеются возможности, то сумеете принять правильное решение.

Заключение

Итак, вы познакомились с сайтом Zen Garden. Что дальше?

Возможно, вы принадлежите к числу ветеранов, пользовавшихся технологией CSS задолго до того, как это обрело смысл. А возможно, вы только сейчас познакомились с языком.

В любом случае, есть шанс, что вы захотите продолжить знакомство с CSS. В этой части книги мы расскажем, как это сделать.

Шпаргалка по CSS

Любому, кто пытался создать хотя бы парочку сайтов с применением CSS, известно, что поддержка со стороны браузеров пока далека от совершенства. Не исключено, что если стоящая перед вами задача трудна, то все кончится битьем головой об стол и клятвами никогда, ни за что на свете не подступаться больше к веб-дизайну, уж лучше заняться чем-нибудь попроще, скажем, городским планированием или ядерной физикой.

Но важно понимать, что вы не одиноки в своих мучениях. Есть целое сообщество людей, занимающихся проблемами дизайна на основе стандартов, и они готовы делиться друг с другом знаниями.

Из многих посвященных этой теме ресурсов упомянем сайт CSS Crib Sheet, который координирует Дейв Ши – один из авторов этой книги. Он был задуман как наиболее полный справочник по типичным проблемам и их решениям, ниже мы приводим размещенные там материалы. Последняя версия сайта находится по адресу www.mezzoblue.com/css/cribsheet.

Если сомневаетесь, проверяйте

Вы можете избавить себя от лишней головной боли во время отладки, если сразу проконтролируете свой код. Некорректно составленные XHTML-разметка и таблицы стилей могут привести к различным проблемам при верстке. Пользуясь инструментами контроля, вы научитесь искать и исправлять ошибки. На сайте W3C имеются программы для проверки XHTML и CSS-кода, доступные соответственно по адресам http://validator.w3.org и http://jigsaw.w3.org/css-validator.

Отметим, что ошибки в начале документа могут быть причиной наведенных ошибок ниже из-за неправильной вложенности или по другим причинам. Мы рекомендуем начать с исправления нескольких очевидных ошибок, а потом повторить проверку, в результате общее число ошибок может значительно сократиться.

Создавайте и тестируйте таблицы стилей сначала в самых передовых браузерах, а затем уже в остальных

Если вы разрабатываете и тестируете сайт в старом браузере, то не исключено, что сайт окажется зависимым от недостаточно развитого и, возможно, некорректно работающего механизма прорисовки. И тогда будет очень обидно видеть, что в браузере, лучше совместимом со стандартами, сайт отображается неправильно.

Лучше отталкиваться от идеала, а *затем* переходить к браузерам с меньшими возможностями. Тогда ваш код будет отвечать стандартам и не придется прилагать гигантских усилий для поддержки других браузеров. На сегодняшний день это означает, что тестировать следует сначала в браузерах Mozilla (Firefox), Safari или Opera, а только потом вносить исправления для Microsoft Internet Explorer.

Используя плавающее позиционирование, проверьте, правильно ли расчищается место

Плавающее позиционирование – это сложная техника, которая не всегда дает ожидаемых результатов. Если оказывается, что плавающий элемент выходит за границы объемлющего элемента или вообще ведет себя не так, как надо, проверьте правильность своих действий. Ознакомьтесь с руководством Эрика Мейера по этому вопросу на сайте Complex Spiral (www.complexspiral.com/publications/containing-floats).

В избежание схлопывания полей задавайте отступы или границы

Иногда приходится бороться с лишним пустым пространством, а иногда оно не появляется там, где вам нужно. Если вы используете поля, то, возможно, причина неприятностей – это схлопывание. Энди Бадд (Andy Budd) на своем сайте объясняет, чего следует ожидать (www.andubudd.com/archives/2003/11/no_margin_for_error).

Старайтесь избегать одновременного задания для элемента отступов/границ и фиксированной ширины или высоты

В Internet Explorer 5.0 и 5.5 для Windows модель прямоугольных областей реализована некорректно, что часто приводит к проблемам при вычислении ширины и высоты. Существуют способы обойти эту ошибку (см. http://tantek.com/CSS/Examples/boxmodelhack.html), но лучше вообще держаться от нее подальше и задавать свойство `margin` для потомков, если у родителя должна быть фиксированная ширина или высота, либо свойство `padding` для родителя, если конкретная ширина/высота должна быть у потомка.

Избегайте эффекта появления нестилизованного контента в Internet Explorer 6 для Windows

Если для импорта внешних стилей вы пользуетесь исключительно директивой `@import`, то рано или поздно обнаружите, что в Internet Explorer 6 для Windows на мгновение появляется простой, неформатированный HTML-код *еще до применения к нему* стилей. Этого можно избежать, присоединяя таблицу стилей по ссылке или поместив на страницу элемент `script`. См. статью «Flash of Unstyled Content» на странице www.bluerobot.com/web/css/fouc.asp.

Не полагайтесь на свойства min-width/min-height

Internet Explorer для Windows не поддерживает ни то, ни другое. Зато свойства `width` и `height` он трактует так, как будто речь идет о `min-width` и `min-height`, поэтому, применив фильтрацию, вы сможете добиться нужного результата с помощью одних лишь `width` и `height`. Конкретный пример см. в статье «Min-height: fixed» на странице mezzoblue.com/archives/2004/09/16/minheight_fi.

Если сомневаетесь, уменьшите значения, выраженные в процентах

Иногда из-за ошибок округления 50% + 50% дает 100,1%, и весь макет плывет. Попробуйте вместо 50% задать 49% или 49,9%.

Если контент неправильно отображается в Internet Explorer

Возможно, вы стали жертвой ошибки «Peekaboo[1] bug». Это, например, происходит, когда пользователь задерживает курсор мыши над ссылкой. О том, как обойти эту ошибку, написано на сайте Position Is Everything (www.positioniseverything.net/explorer/peekaboo.html).

[1] Peekaboo – это игра, в которую взрослые играют с маленькими детьми. Смысл в том, что вы закрываете лицо руками, а потом внезапно убираете руки с возгласом «Boo!». Суть ошибки в том, что при определенных действиях пользователя и определенной разметке контент то исчезает, то появляется. – *Прим. пер.*

Проверьте, а возможен ли вообще нужный вам эффект

Некоторые браузеры поддерживают расширения CSS, отсутствующие в официальной спецификации. Так, фильтры и форматирование полосы прокрутки работают только в Internet Explorer. Если программа контроля говорит, что используемая вами конструкция не определена, то, скорее всего, речь идет о нестандартном расширении, которое не будет поддержано другими браузерами.

Не забывайте о порядке LoVe/HAte

Псевдоклассы для ссылок всегда нужно задавать в следующем порядке: link, visited, hover, active. При любом другом порядке конечный результат не гарантирован. Если вы пользуетесь также псевдоклассом `:focus`, то применяйте порядок LVHFA (Мэтт Хоуи предложил такую мнемоническую фразу «Lord Vader's Handle Formely Anakin»).

Не забывайте о порядке следования значений в стенографических свойствах

Стенографические свойства `border`, `margin` и `padding` подразумевают определенный порядок следования значений: против часовой стрелки, то есть top, right, bottom, left. Следовательно, запись `margin: 0 1px 3px 5px;` означает, что верхнее поле отсутствует, правое равно 1 пикселю и так далее. Запомните мнемоническое слово «TRouBLe», и вы всегда будете указывать порядок правильно.

Указывайте единицы измерения для ненулевых значений

Спецификация CSS настаивает на указании единиц измерения для кегля шрифтов, величин полей и других линейных размеров. (Единственное исключение составляет свойство `line-height`, для которого задавать единицу измерения необязательно, хотя и допустимо.) Не следует полагаться на поведение конкретного браузера в случае отсутствия единиц измерения. Впрочем, нуль – всегда нуль, будь он выражен в `px`, `em` или иных единицах, поэтому разрешается не задавать единицу измерения для значения 0. Пример: `padding: 0 2px 0 1em;`

Проверьте, как выглядит страница при разных размерах шрифтов

Передовые браузеры, к примеру Mozilla и Opera, позволяют пользователю изменять размер текста вне зависимости от единицы измерения. У некоторых пользователей наверняка выставлен иной размер шрифта по умолчанию, нежели у вас. Старайтесь, чтобы ваша верстка выглядела пристойно в максимально широком диапазоне размеров шрифтов.

Следите за соответствием регистра в HTML и CSS

Согласно спецификации, язык CSS чувствителен к регистру букв, и большинство браузеров следуют этому требованию. Имя `'homePage'` для класса вполне допустимо при условии, что оно записано одинаково в HTML-разметке и в таблицах стилей. Применение же правил к элементам класса `'homepage'` в этих условиях приведет к ошибкам в пользовательских агентах, которые строго соблюдают спецификацию (например, в Mozilla).

Тестируйте на встроенных таблицах стилей, эксплуатируйте на импортированных

Если на этапе разработки вы помещаете таблицу стилей между тегами `<style>` и `</style>`, то устраняете потенциальную причину ошибок из-за кэширования. При работе с некоторыми старыми браузерами это бывает весьма существенно. Но не забывайте переносить весь код CSS во внешний файл перед запуском в эксплуатацию. Этот файл можно импортировать (`@import`) или связать (`<link>`). В противном случае преимущества внешних таблиц стилей будут утрачены.

Добавляйте видимые рамки на этапе отладки верстки

Универсальное правило типа `* {border: solid 1px #f00;}` может сильно помочь при поиске ошибок в макете. Добавление рамки к отдельным элементам поможет выявить перекрытия и лишнее пустое пространство, которое иначе было бы трудно заметить.

Не заключайте путь к изображениям в одиночные кавычки

При указании пути к фоновым изображениям кавычки вообще необязательны, но если он заключен в одиночные кавычки, то Internet Explorer 5 для Macintosh не загружает картинку. Пользуйтесь либо двойными кавычками, либо обходитесь без них.

Будьте аккуратны при стилизации ссылок на внутренние анкеры

Если в вашем коде встречается классический анкер – `<a name="anchor">`, то к нему будут применяться псевдоклассы `:hover` и `:active`. Чтобы этого избежать, пользуйтесь для анкеров атрибутом `id` – `<div id="anchor">` или применяйте несколько более длинную синтаксическую конструкцию – `:link:hover`, `:link:active`. См. www.dbaron.org/css/1999/09/links.

Не ссылайтесь на пустые файлы, резервируя место для будущих таблиц стилей (например, для КПК или печатных устройств)

Браузер Internet Explorer 5 для Mac некорректно обрабатывает пустые таблицы стилей, из-за чего время загрузки страницы возрастает. Поместите в файл хотя бы одно правило (можно даже комментарий) или просто уберите ссылку на него.

Поймите суть проблемы

Быть может, самый важный наш совет состоит в том, что уяснение сути возникшей проблемы – это ключ к ее решению. Понимание приходит с опытом, после многих часов, потраченных на составление кода.

Предположим, что вы заметили пустую горизонтальную полоску высотой 3 пикселя перед плавающим элементом в браузере Internet Explorer для Windows, и только в нем. Если вы не знаете, что в этом браузере есть ошибка, проявляющаяся при работе с плавающими элементами, то будете пробовать отрицательные поля, относительное позиционирование, а возможно, даже какие-то трюки, чтобы подправить положение окружающих элементов и избавиться от ненужного промежутка.

Но, зная заранее об уникальной для Internet Explorer проблеме, заключающейся в увеличении полей некоторых плавающих элементов точно на три пикселя, вы не станете гадать и пробовать все вышеперечисленные способы.

Попытка исправить необъяснимые ошибки приводит к их накоплению. Применив на странице несколько обходных маневров, вы создаете зависимость от имеющихся в конкретном браузере ошибок алгоритма прорисовки. Отладка верстки на более поздних этапах дизайна лишь усложнится, если сразу не решить проблем, возникших в начале работы.

Понимая, что же на самом деле послужило причиной ошибки, вы сможете подойти к ее исправлению более осмысленно. Сайт Crib Sheet – это только отправная точка. Существует еще множество сайтов, на которых детально рассматриваются дефекты, имеющиеся в разных браузерах. Мы настоятельно рекомендуем заглянуть на великолепный сайт Position Is Everything (www.positioniseverything.net).

(Кстати говоря, если вас заинтересовала ошибка, связанная с тремя пикселями, то на этом сайте вы узнаете, как исправить ее, а заодно и много других специфичных для Internet Explorer ошибок, см. страницу www.positioniseverything.net/articles/hollyhack.html#haslayout).

Другие ресурсы

Мы с удовольствием упомянем несколько замечательных сайтов и периодических изданий. Кроме них, есть немало ресурсов, посвященных CSS и стандартам, надо только знать, где искать.

Сайты о дизайне с применением CSS

CSS Discuss

www.css-discuss.org

Объемный список рассылки с новостями из мира CSS и практическими советами. Здесь спрашивают и отвечают как новички, так и профессионалы.

CSS на сайте MaxDesign

http://css.maxdesign.com.au

Сборник подробных, узкоспециальных статей о списках, плавающих элементах и многом другом.

Подборка ссылок по CSS и стандартам Web

www.andybudd.com/links/cssweb_standards

Постоянно обновляемая подборка ссылок на статьи по CSS, описание различных приемов и советы по исправлению ошибок. Сайт поддерживает Энди Бадд.

Holy CSS Zeldman!

www.dezwozhere.com/links.html

Эндрю Фернандес составил, наверное, самую большую и хорошую подборку ссылок на различные онлайновые ресурсы, посвященные CSS.

Position Is Everything

www.positioniseverything.net

У вас проблема с CSS? Нашли новую ошибку? Загляните на сайт Большого Джона и Холли – наверняка они утолят ваши печали.

Публикации

A List Apart

www.alistapart.com

Давно издающийся журнал для веб-дизайнеров.

Digital Web Magazine

www.digital-web.com

Обзоры книг и программ для веб-дизайнера, а также многое другое.

Web Standards Project Buzz

www.webstandards.org/buzz

Постоянно обновляемая информация обо всем, что касается стандартов Web.

Книги

Рекомендуем следующие книги для начинающих и профессионалов:

1. Eric Meyer. Cascading Style Sheets: The Definitive Guide. O'Reilly, 2000.
2. Eric Meyer on CSS: Mastering the Language of Web Design. New Riders, 2002.
3. More Eric Meyer on CSS. New Riders, 2004.
4. Jeffrey Zeldman. Designing with Web Standards. New Riders, 2003 («Web-дизайн по стандартам», «НТ Пресс», 2005).
5. Molly Holzschlag. Special Edition: Using XHTML and HTML. Que, Special Edition 2002.

Ждем вашу работу!

Сайт Zen Garden останется открытым для желающих прислать свои работы и после выхода этой книги. Изучив представленные образцы, почему бы не предложить собственный вариант дизайна? (Не забудьте упомянуть о книге, когда будете посылать свое творение!)

Все новые поступления тщательно оцениваются по нескольким показателям, изменяющимся от 1 до 37, а затем предлагаются на суд дрессированных обезьян... А может, и нет. На самом деле, решение бинарное – либо палец вверх, либо вниз – не сложнее.

Просим понять, что не все присланные работы попадают в галерею, это вопрос соотношения количества и качества. Достойными считаются работы, содержащие нестандартные и тщательно отделанные визуальные образы, на интересную тему или составленные с юмором, и в которых применяются принципы правильной верстки и типографии. Есть еще иные факторы, которые трудно выразить словами. Лучший совет, который мы можем дать, – сравните свою работу с теми, что имеются в официальном списке и, если она не уступает уже представленным высококачественным образцам, присылайте.

Оценка и добавление новых работ в галерею производятся тогда, когда это удобно Дэйву Ши, поэтому наберитесь терпения: рано или поздно вы получите извещение о том, принята ли ваша работа.

Предметный указатель

Символы

А

Б

В

Г

Д

Е

З

И

К

Л

М

Т

Ф

D

E

F

G

H

I

J

L

M

Дэйв Ши
Молли Е. Хольцшлаг

Философия CSS-дизайна

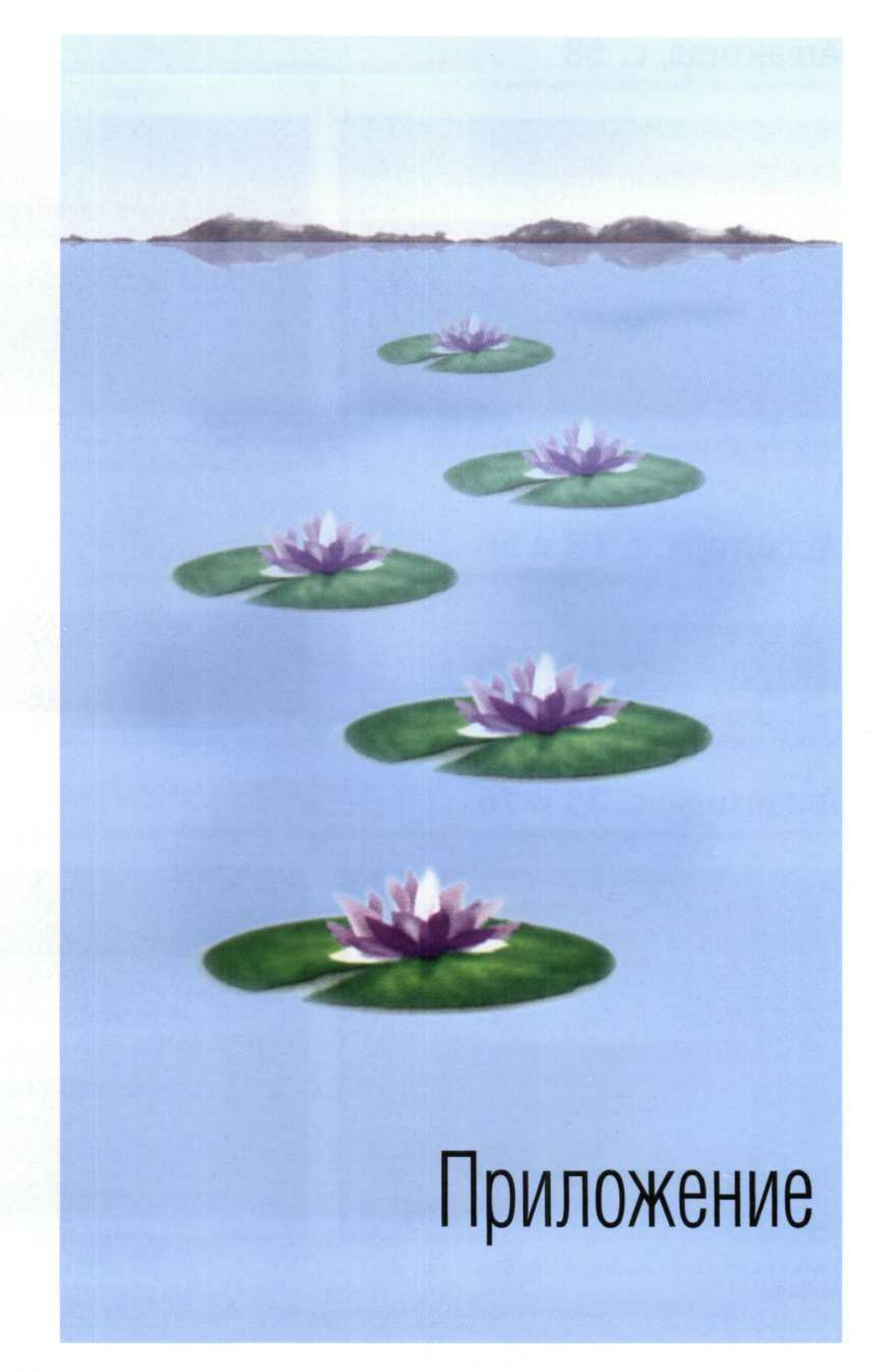
Научный редактор *Слинкин А. А.*
Ответственный редактор *Теренина О. А.*
Верстка *Кабанов В. В.*
Графика *Шахина А. Г.*
Дизайн обложки *Харевская И. А.*

Издательство «НТ Пресс», 129085, Москва,
Звездный б-р, д. 21, стр. 1.

Издание осуществлено при техническом участии
ООО «Издательство АСТ»

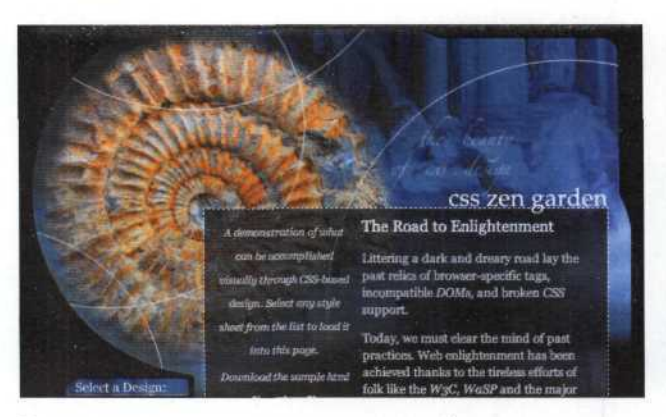
Издано при участии ООО «Харвест». Лицензия № 02330/0056935 от 30.04.04.
РБ, 220013, Минск, ул. Кульман, д. 1, корп. 3, эт. 4, к. 42.

Республиканское унитарное предприятие
«Издательство «Белорусский Дом печати».
220013, Минск, пр. Независимости, 79.



Приложение

Атлантида, с. 58



Оранжевые тона раковины создают цветовой контраст



Заголовок в одних лишь синих тонах больше соответствует теме, но ему недостает контрастности

Атлантида, с. 68 и 69

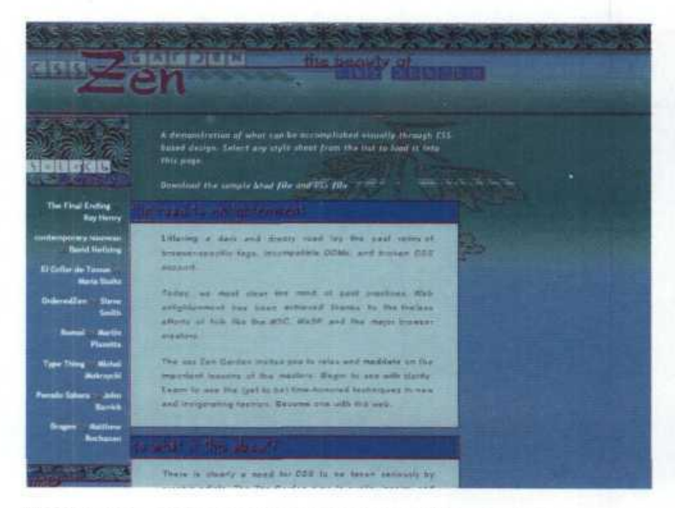


Палитра цветов дизайна «Весна»
включает оттенки зеленого, намек на голубой и белый цвет

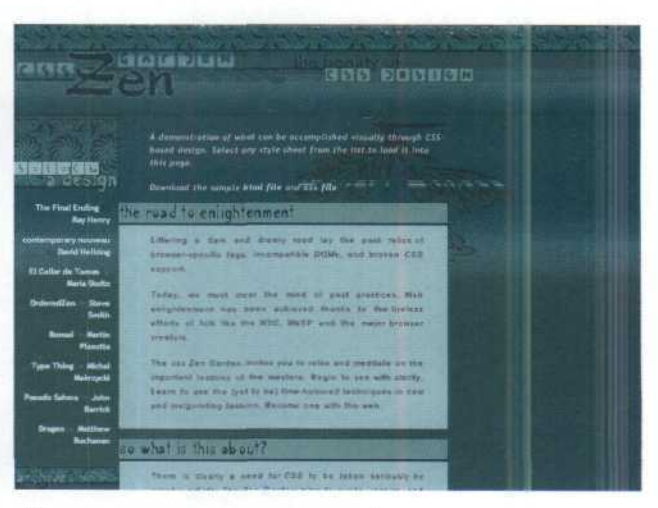


Создание цветовой палитры в Photoshop

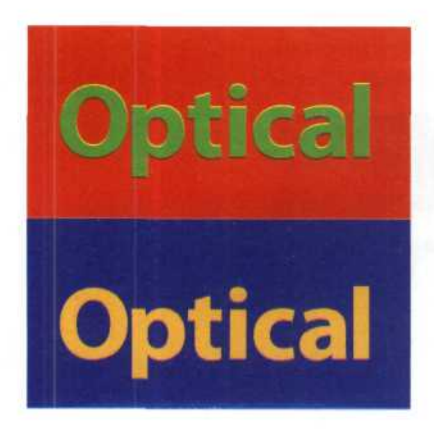
Атлантида, с. 75 и 76



Увеличение контрастности приводит к конфликту с другими деталями



«В зеленых тонах» при недостаточной контрастности

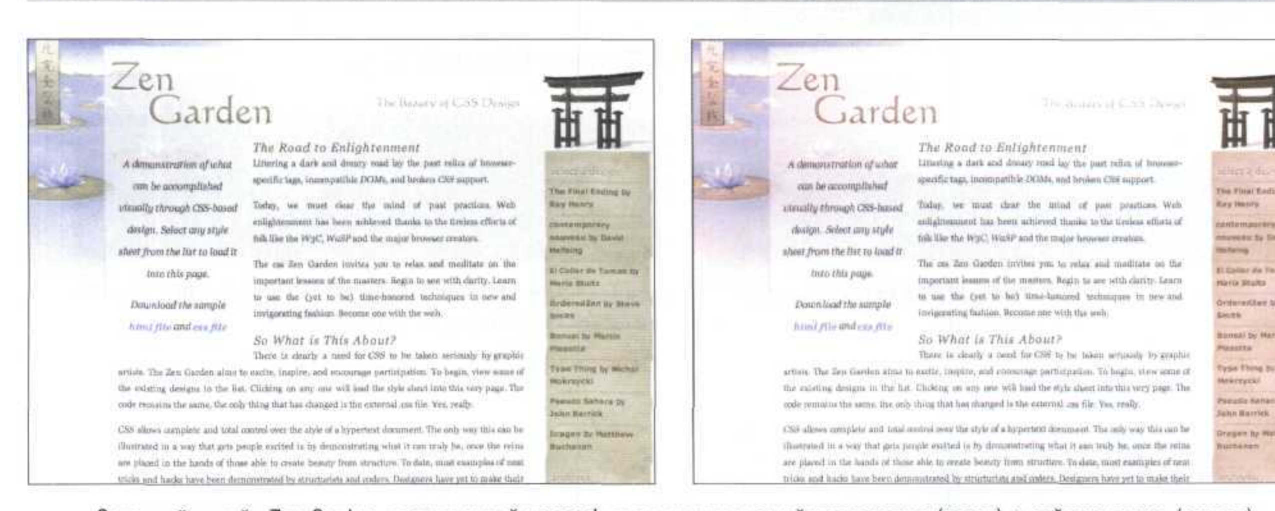


Одновременный контраст приводит к иллюзии мерцания на границах основных цветов



Один и тот же цвет кажется более оранжевым на желтом фоне и более желтым на красном

Атлантида, с. 77



Основной дизайн Zen Garden, пропущенный через фильтр, имитирующий протанопию (слева) и дейтеранопию (справа)

Японский сад, с. 134 и 135



Изображение сохранено в формате GIF с 256 цветами, размер файла равен 8,6 Кб



Изображение сохранено в формате GIF с 32 цветами, размер файла равен 5,0 Кб



Изображение сохранено в формате GIF с 4 цветами, размер файла равен 2,1 Кб



Изображение сохранено в формате JPEG с коэффициентом качества 70, размер файла равен 8,3 Кб



Изображение сохранено в формате JPEG с коэффициентом качества 30, размер файла равен 3,7 Кб



Изображение сохранено в формате JPEG с коэффициентом качества 10, размер файла равен 2,5 Кб



Изображение сохранено в формате PNG с глубиной цвета 24 бит, размер файла равен 16,2 Кб



Изображение сохранено в формате PNG с глубиной цвета 8 бит и 64 цветами, размер файла равен 5,3 Кб



Изображение сохранено в формате PNG с глубиной цвета 8 бит и 4 цветами, размер файла равен 2,2 Кб

Японский сад, с. 136



Прозрачный GIF, который был подготовлен для белого фона, отображается на белом фоне

Прозрачный GIF, который был подготовлен для белого фона, отображается на светло-зеленом фоне

Прозрачный GIF, который был подготовлен для белого фона, отображается на темно-зеленом фоне

Береговой бриз, с. 161



Рис. 6 ✦ Исходная поверхность листа



Для имитации прожилок и текстуры добавлены тени и выделения

Два океана, с. 179



Контрастность шрифта в заголовке «Двух океанов»

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Инверсные цвета в тексте, расположенном ниже заголовка

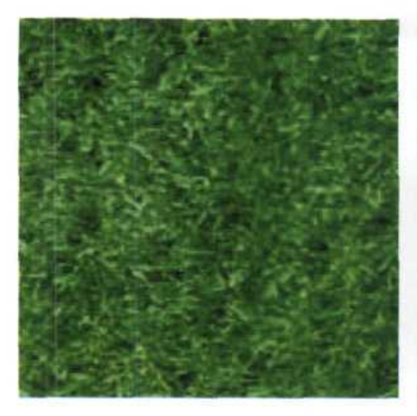
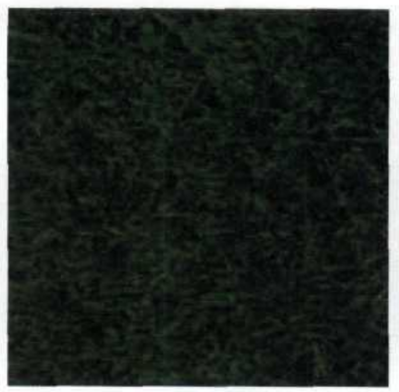
Жажда крови, с. 202



После размывания становится видно, что у блоков текста разные оттенки

Примеры разного цвета шрифта

Упругая лужайка, с. 250

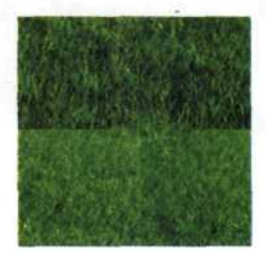

Изображения травы, которыми мостится «Упругая лужайка»: ярко-зеленое для основного фона, коричневое для боковой колонки и темно-зеленое для главной области контента



При повторении видны стыки



На одной смещенной плитке видно несоответствие цветов на стыках



В результате клонирования примыкающих областей шов закрашивается

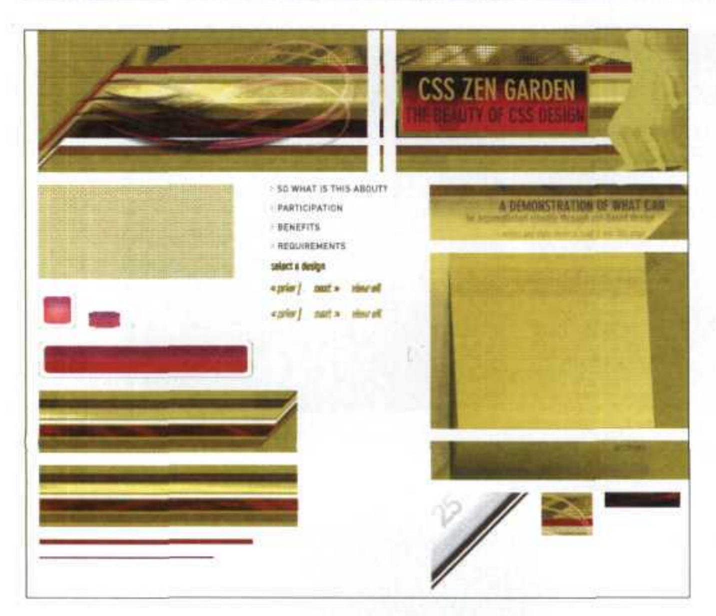


Окончательный вид плитки



Теперь стыки гораздо менее заметны

Мнемоника, с. 285



Все изображения, использованные в «Мнемонике»



css Zen Garden

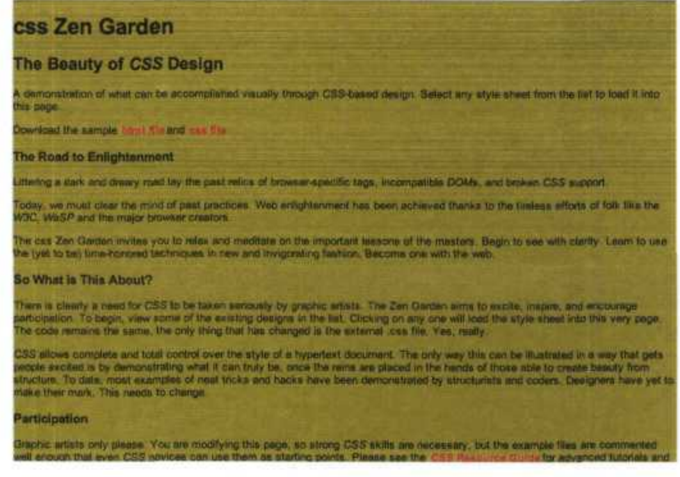
The Beauty of CSS Design

The Road to Enlightenment

So What is This About?

Participation

Вид страницы после применения начальных стилей, задающих цвета ссылок и шрифты



css Zen Garden

The Beauty of *CSS* Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the *W3C*, *WaSP* and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for *CSS* to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong *CSS* skills are necessary, but the example files are commented well enough that even *CSS* novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and

Первая фоновая картинка присоединена к элементу `body`